

С. Апунович, В. Бойко, Г. Злобін, С. Кудрик, В. Семенюк  
під редакцією Г. Злобіна

# **Linux – це просто як Borsch**

(шкільний проект)

Львів 2006



У цьому посібнику розглянуті методи роботи з ОС **Linux** на базі інсталяційної збірки **Borsch**. Збірка створена на основі **Debian GNU/Linux**. Відбір програм здійснювався з метою використання цієї збірки у шкільному курсі інформатики на ПЕОМ з процесором **Intel Celeron 466** і 128 Мб оперативної пам'яті. Розглянуті методи роботи з графічними оболонками **XFCE**, **Gnome** і **KDE**. Подана інформація про офісні пакети **OpenOffice.org1.1.0** і **GNOME Office**. Розглянуті методи роботи з комп'ютерними словниками, педагогічними програмними продуктами а також виконання **MS Window**-програм за допомогою системи **Wine**. Викладено основи адміністрування ОС **Linux**.

Для вчителів і викладачів інформатики.

Відомості про авторів:

Степан Апуневич, науковий співробітник астрономічної обсерваторії Львівського національного університету імені Івана Франка, кандидат фізико-математичних наук. Автор розділів 1.1, 1.5, 1.6.1 – 1.6.8, 2.3.3;

Василь Бойко, студент другого курсу факультету електроніки Львівського національного університету імені Івана Франка. Автор розділу 1.6.9;

Григорій Злобін, доцент кафедри радіофізики Львівського національного університету імені Івана Франка, кандидат технічних наук. Автор розділів 1.2 – 1.4, 2.1 – 2.4 (за винятком 2.3.3.), 2.5.7, 2.5.8, 2.7.3;

Сергій Кудрик, інженер. Автор українізації офісних пакетів OpenOfficeorg 1.1, OpenOfficeorg 2.0;

Валерій Семенюк, вчитель інформатики Львівської СШ№80, методист Львівського обласного інституту освіти. Автор розділу 2.6.

Права авторів застережено.

Підготовлено до друку в офісному пакеті **OpenOfficeorg1.1.0** в ОС **Linux**

Зміст	
Передмова	5
1. Операційна система <b>Linux</b>	7
1.1. Історія ОС <b>Linux</b>	7
1.2. Файлова система ОС <b>Linux</b>	8
1.3. Розпочинаємо працювати з <b>Linux</b>	10
1.4. Графічний інтерфейс в ОС <b>Linux</b>	11
1.4.1. Графічна оболонка <b>XFCE</b>	11
1.4.2. Графічна оболонка <b>GNOME</b>	17
1.4.3. Графічна оболонка <b>KDE</b>	21
1.5. Командний рядок в ОС <b>Linux</b>	25
1.6. Адміністрування системи ОС <b>Linux</b>	30
1.6.1. Зміст поняття ОС <b>Linux</b>	30
1.6.2. Структура ОС <b>Linux</b>	31
1.6.3. Файлова система ОС <b>Linux</b>	32
1.6.4. Використання накопичувачів інформації	36
1.6.5. Від увімкнення до роботи	38
1.6.6. Управління користувачами	39
1.6.7. Встановлення обладнання в ОС <b>Linux</b>	41
1.6.8. Методи встановлення програм в ОС <b>Linux</b>	42
1.6.9. Клонування <b>Debian GNU/Linux</b>	45
2. Прикладне програмне забезпечення для IBM-подібних ПЕОМ	48
2.1. Офісний пакет <b>OpenOffice.org</b>	48
2.1.1. Текстовий процесор <b>OpenOffice.orgWriter</b>	49
2.1.1.1. Уведення тексту	50
2.1.1.2. Збереження тексту на магнітних дисках та відкриття існуючого текстового файлу	53
2.1.1.3. Друк текстового документа	53
2.1.1.4. Пошук заданої інформації	55
2.1.1.5. Робота з таблицями	56
2.1.1.6. Робота з графікою	56
2.1.2. Табличний процесор <b>OpenOffice.org.Calc</b>	59
2.1.3. Редактор векторної графіки <b>OpenOffice.org.Draw</b>	64
2.1.4. Система побудови презентацій <b>OpenOffice.org.Impress</b>	65
2.2. Офісний пакет <b>GNOME Office</b>	68
2.2.1. Текстовий процесор <b>Abiword</b>	68
2.2.1.1. Уведення тексту в ПЕОМ	68
2.2.1.2. Збереження тексту на магнітних дисках	70
2.2.1.3. Друк текстового документа	70
2.2.1.4. Пошук заданої інформації	71
2.2.1.5. Робота з таблицями	72
2.2.1.6. Робота з графікою	73
2.2.2. Табличний процесор <b>Gnumeric</b>	73
2.3. Опрацювання графічної інформації	78
2.3.1. Графічний редактор <b>KolourPaint</b>	78
2.3.2. Графічний редактор <b>Gimp</b>	79
2.3.3. Векторний графічний редактор <b>InkScape</b>	80
2.4. Машинний переклад тексту	82
2.5. Програмування в ОС <b>Linux</b>	84
2.5.1. Історичний спадок	84

2.5.2. Засоби	83
2.5.3. Бібліотеки	85
2.5.4. Утиліти	86
2.5.5. Інтегровані системи розробки	86
2.5.6. Інтегроване середовище розробки <b>Anjuta</b>	87
2.5.7. Оболонка <b>Free Pascal</b> для програмування мовою Паскаль	88
2.5.8. Оболонка <b>Algo</b> для програмування мовою Паскаль	89
2.6. Педагогічні програмні засоби в ОС <b>Linux</b>	91
2.7. Глобальна мережа Інтернет і засоби для роботи у ній	98
2.7.1. Обладнання для під'єднання до глобальних мереж	99
2.7.2. Основні служби Інтернету	102
2.7.3. Засоби ОС <b>Linux</b> для роботи в Інтернет	103
2.8. Системи керування базами даних (СКБД)	104
2.8.1. Правознавча система “ <b>Національні акти України</b> ”	105
3. Завдання до практичних занять з шкільного курсу інформатики	107
Завдання I. Початок і закінчення роботи в ОС <b>Linux</b>	107
Завдання II. Дії з файлами і каталогами в ОС <b>Linux</b>	108
Завдання III. Методи запуску програм та маніпулювання вікнами програм	108
Завдання IV. Опрацювання текстової інформації на ПЕОМ	108
Завдання V. Опрацювання числової інформації на ПЕОМ	109
Завдання VI. Опрацювання графічної інформації на ПЕОМ	116
Завдання VII. Пошук інформації у глобальній мережі Інтернет і робота з електронною поштою	116
Список рекомендованої літератури	117

## Передмова

За переказами у 1976 р. Стефан Возняк та Стівен Джобс в автомобільному гаражі під яблуною зібрали з мікросхем телевізійний ігровий пристрій з програмним керуванням - зміст та форма гри визначались програмою, яка завантажувалась в оперативну пам'ять пристрою. Програмісти, яких винахідники попросили оцінити "скоєне", поздоровили винахідників з створенням першого в світі персонального комп'ютера (англ. **personal computer** - персональний обчислювач. На думку авторів в українській мові більш доречним є термін **персональна електронна обчислювальна машина - ПЕОМ**). Окрилені такою оцінкою, винахідники створили фірму **Apple**, яка за кілька років вийшла на рівень мільйонних продаж своїх ПЕОМ у США. Простота користування персональним комп'ютером привела до небаченого в історії США явища - службовці за свої кошти купували ПЕОМ для використання їх на роботі. Фірма **IBM**, яка спочатку спогорда спостерігала "дитячі ігри" з маленькими комп'ютериками **Apple**, раптом відчула, що на ринку з'явився серйозний конкурент. І у 1981 році невеличка група інженерів фірми **IBM** виставила на загальний огляд **IBM PC** з дуже скромними характеристиками (процесор I8088 - 16-бітний з 8-бітною шиною даних, 64К оперативної пам'яті, вбудований Бейсік, кольоровий телевізор замість дисплея). Архітектура **IBM PC** була оголошена відкритою - фірма **IBM** сподівалася у такий спосіб заощадити кошти на розробці периферійних пристроїв до **IBM PC** - незалежні виробники, маючи документацію про архітектуру **IBM PC**, могли незалежно від фірми **IBM** (і за власні кошти) розробляти пристрої друку, дисплеї, пристрої читання/запису інформації на магнітні диски. Однак, попри очікуваний ефект, фірма **IBM** отримала несподіваний результат - незалежні виробники, користуючись відкритою архітектурою **IBM PC**, почали виробляти **IBM**-подібні ПЕОМ. Певні зміни в архітектурі **IBM**-подібної ПЕОМ, які робилися для того, щоб уникнути судових переслідувань з боку фірми **IBM**, узгоджувались з **BIOS** (базовою системою вводу-виводу), і для користувача ПЕОМ ставала дуже подібною на **IBM PC**. Скромні можливості **IBM PC** (а отже і низька конкурентноздатність) змусили фірму **IBM** незабаром виставити **IBM PC XT** (**eXtended arhiTecture**) та **IBM PC AT** (**Advanced arhiTecture**). Ці ПЕОМ уже мали звичний для нас вигляд - системний блок з накопичувачами на ГМД та ЖМД, дисплей, клавіатура, мишка. Однак вгнатися за конкурентами, які "осіддали" відкритість архітектури **IBM PC**, фірма **IBM** вже не могла. Не отримавши очікуваного результату від відкритості архітектури **IBM PC**, фірма **IBM** спробувала закрити архітектуру своєї нової лінії **IBM PS/2** (до речі усі ПЕОМ фірми **Apple** мають закриту архітектуру), однак очікуваного результату це також не дало - світовий ринок вже був наповнений дешевшими **IBM**-подібними ПЕОМ з співмірними або й кращими характеристиками. Skorистався з відкритості архітектури **IBM PC** і Радянський Союз - у другій половині вісімдесятих років були розроблені ПЕОМ **ЕС-1840** (Мінськ), **Іскра-1030** (Смоленськ), **Нейрон** (Київ). Багатоплатна конструкція цих ПЕОМ стала причиною їх низької надійності. Особливе місце серед цих розробок займають "**Поиск-1**" (практично повна копія **IBM PC**), "**Поиск-2**" (копія **IBM PC XT**), які вироблялись на Київському виробничому об'єднанні "Електронмаш", та "**Практик**", "**ЕС 7978**" (копія **IBM PC XT**), які вироблялись на Канівському електромеханічному заводі "Магніт". Відмічаючи українські розробки, варто згадати ще дві події:

І. На початку 60-х років минулого століття у Київському інституті кібернетики АН УРСР була розроблена ЕОМ **Мир-2**, яка принципово орієнтувалась на *персонального користувача*, мала графічний дисплей та "світлове перо", могла виконувати аналітичні обчислення (вперше у світі). Дві! такі ЕОМ були продані у США. Однак, оскільки така розробка йшла врозріз з вказівками ЦК КПРС про розробки у СРСР радянських копій лише двох ліній ЕОМ - **IBM360** (серія **ЕС10xx**) та **PDP11** (серія **СМ-х**), подальші роботи були припинені. А даремно!

II. У 1976 році професор В. Петров (інститут фізики АН УРСР) на Всесвітньому електротехнічному конгресі виголосив доповідь “Оптичний диск як універсальний носій інформації”. І знову пішли заперечення - “Навіщо ви це робите? Американці ж у цьому напрямку не працюють. І нам не треба.”. Після випуску 16 систем оптичної пам’яті для великих ЕОМ виробництво у Кам’янці-Подільському було припинено.

Крім **IBM**-подібних ПЕОМ та ПЕОМ фірми **Apple** випускались персональні ЕОМ і інших ліній, слід згадати **Commodor** (фірми **Amiga**), **Atari** (фірми **Atari**), **ZX-Spectrum** (фірми **Zilog** та його радіолюбительські аналоги, зроблені у Львові), **ПК-Львів** (розробник В.Пуйда), **Правець-8**, **БК-0010**, **Корвет**, **Агат**. Усі перераховані ПЕОМ випускались в одному корпусі з клавіатурою, використовували 8-бітні мікропроцесори різних виробників і були непристосовані до модернізації, саме тому з появою більш потужних мікропроцесорів вони просто зійшли з арени.

Сучасні ПЕОМ можна розділити на дві групи:

персональні ЕОМ від фірми **Apple (Macintosh)**;

**IBM**-подібні ПЕОМ від різних виробників ( у тому числі українських).

Персональні ЕОМ від фірми **Apple (Macintosh)** завжди відрізнялись високими споживчими характеристиками, які забезпечувались високим рівнем схемних рішень та високою якістю програмного забезпечення, авторам доводилось спостерігати у роботі **MAC OS5** з графічним інтерфейсом, яка вантажилась з 3,5” дискети (порівняйте 1,44Мб з мінімумом у 100Мб для **Microsoft Windows 95**). Політика закритої архітектури (донедавна **Apple** та **Macintosh** випускали лише 4 заводи в світі) дозволяла випускати ПЕОМ лише високої якості. Розробники програмного забезпечення, які бажають писати програми для **Apple**, повинні пройти обов’язкову сертифікацію (у такий спосіб забезпечується високий рівень програмного забезпечення для **Apple**). Однак за високу якість доводиться розплачуватись високою ціною - ціни на ПЕОМ від **Apple** у 2-3 рази вищі від цін співмірних по потужності **IBM**-подібних ПЕОМ. В Україні ПЕОМ **Apple** використовують, як правило, у видавництвах ( якщо видавництво спроможне придбати таку дорогу техніку).

**IBM**-подібні ПЕОМ сьогодні в Україні займають понад 90% парку персональних ЕОМ. Це зумовлено більш низькою ціною, відкритістю архітектури та простотою модернізації і технічного обслуговування (особливо ПЕОМ, зібраних українськими виробниками). Принцип “відкритої архітектури **IBM PC**” у Радянському Союзі, а після його розвалу у країнах СНД, безпідставно був перенесений на програмне забезпечення - комерційне програмне забезпечення, як правило, встановлювалось з піратських оптичних дисків. Внаслідок цього в Україні операційні системи **MS DOS** і **MS Windows** та офісний пакет **MS Office** стали стандартом “де-факто” (а вітчизняні розробки програмного забезпечення майже не велися). До певного часу така поведінка України не призводила до санкцій з боку міжнародної спільноти і США зокрема. Однак з травня 2001 р. розпочалося “наведення порядку” у галузі використання системного і прикладного програмного забезпечення. Стихійна орієнтація на програмне забезпечення від фірм **Microsoft**, **Adobe**, **Borland** і ін. складає серйозну загрозу для України, оскільки навіть з врахуванням пільгової ціни для навчальних закладів вартість програмного забезпечення однієї ПЕОМ складає від 300 до 5000 умовних одиниць. Спроби Міністерства освіти і науки України забезпечити заклади освіти ліцензійним програмним забезпеченням фірми **Microsoft** залишаються утопічними, оскільки оплачується, як правило, лише операційна система **MS Windows**, решта програмного забезпечення залишається неліцензійним. Викликають здивування спроби псевдофахівців закріпити в якості стандарту “де-факто” офісний пакет від фірми **Microsoft**, при цьому повністю ігнорується факт існування офісних пакетів від інших виробників ( **Corel Office**, **Lotus Smartsuite**, **Star Office** і ін.). Використання лише комерційного програмного забезпечення у вищих навчальних закладах небезпечно ще й тим, що закритість коду комерційного ПЗ не дає змоги “*подивитись, як це написано*”. Грубо

кажучи, українським комп'ютерним фахівцям відводиться роль тупого споживача (згадайте серію видань “*для чайників – for dummies [дослівно – для тупакієф]*”). Натомість використання відкритого програмного забезпечення не тільки дозволить українським комп'ютерним фахівцям “*подивитись, як це написано*”, а й увійти у світову спільноту розробників найсучаснішого програмного забезпечення (системного і прикладного).

## 1. Операційна система **Linux**

### 1.1 Історія ОС **Linux** та вільного програмного забезпечення

Історія **Linux** почалася раніше аніж з'явився сам термін. **Linux** – член родини операційних систем **Unix**. Саме поняття операційної системи кристалізувалося на початку 70-тих років під час розбудови систем саме цієї родини. За цей час родина таких систем суттєво розрослась, охопила різні апаратні платформи, навіть встигла пережити судові чвари між компаніями-виробниками. У свій час докладалося багато зусиль для стандартизації ОС такого типу, що вилилося у створення стандарту **POSIX**. Вплив **Unix** як стандарту для операційних систем важко переоцінити. Так чи інакше основні концепції **Unix** запозичені до практично кожної із сучасних операційних систем. Системи **Unix** і досі переважають на ринку серверів, і події останніх років вказують на те, що саме **Linux** разом із своїми “родичами” потрохи наступають на сегмент настільних систем загального користування. **Linux** зародився з потреби створення безплатної **Unix**-подібної операційної системи для **IBM**-подібних ПЕОМ. Проект розпочався у 1991 році у сприятливому середовищі академічних установ, університетів, професійних програмістів та системних адміністраторів, а головним засобом спілкування і координування численних його розробників стала Всесвітня мережа – Інтернет. Головним розробником цієї системи був (і залишається) фінський програміст Лінус Торвальдс, і систему охрестили назвою, виведеною із його імені і слова **Unix**. Від своїх творців ця ОС перейняла спрямованість у першу чергу на ефективність роботи, гнучкість та потужність, а не на примхи ринку чи рекламні потреби. На той момент вже була створена спільнотою сукупність “вільних” засобів розробки та керування, і бракувало лише ядра операційної системи. Саме цю прогалину взявся заповнити тоді ще студент Лінус Торвальдс, щоб таким чином розважитися.

Так чи інакше, нам не уникнути порівняння ОС **Linux** з операційними системами серії **Microsoft Windows**. Внаслідок монополії корпорації **Microsoft** на ринку настільних комп'ютерних систем, майже кожний із вас знайомий з цими продуктами. Крім технічних відмінностей між ОС родини **MS Windows** і **Linux**, між ними існує більш суттєва різниця в ідеологічному плані, культурному підґрунті та економічній основі. Ця різниця зумовила відмінність від продуктів корпорації **Microsoft**, особливо у психологічному сприйнятті самої системи. Однак такі психологічні проблеми характерні радше для просторів СНД, де практично відсутній шар субкультури користувачів **Unix**, який добре розвинутий на Заході, особливо в промисловому та академічному середовищі.

Так сталося, що **Linux** став на стрижні конкурентної боротьби з монополістом на ринку програмного забезпечення для настільних систем, хоча ніколи його розробники не ставили собі такої мети, адже вони створювали **Unix**, а не аналог **MS Windows**. Факт того, що він витримав шалений тиск, створений наддержавною за потужністю корпорацією, стверджує життєздатність самої моделі його створення. Ця модель базується на особливому трактуванні авторського права, і на юридичних документах, що формулюють ці концепції, вільних (відкритих) ліцензіях, наприклад **GPL**, громадська ліцензія **GNU** (англ. **GNU Public License**). Ця ліцензія дозволяє вільне копіювання, перенесення і навіть вільну модифікацію програмного забезпечення за умови непривласнення авторського чи суміжних прав на нього з метою вимагання ліцензійних виплат. Іншими словами, коли ви створюєте програму, і під

час її створення використовуєте частини із програм (це не стосується бібліотек, для них створено окремих підвид ліцензії), що поширюються на умовах **GPL**, то створену програму вам теж слід опублікувати і зробити вільнодоступною на тих же умовах, а отже і безоплатною.

Слід розуміти, що не безплатність є основною метою такого програмного забезпечення, а вільність. **Linux** – не єдиний представник вільного програмного забезпечення, повний перелік таких програм містить 10000 пунктів. Не стоять на місці й інші проекти із створення вільної операційної системи **GNU Hurd**, **Plan 9**, **Inferno**.

Задля повноти слід зауважити, що не всі “вільні” програми та операційні системи поширюються на умовах саме громадської ліцензії **GNU**, існує ще декілька видів вільних ліцензій, що, наприклад, менш суворі стосовно непривласнення. Тут слід згадати операційні системи **BSD**, що керуються відповідною ліцензією, а саме **Unix FreeBSD**, **OpenBSD**, **NetBSD**. У певному сенсі до цієї родини можна зарахувати ОС **MacOS X** виробництва корпорації **Apple**, яка базується на ядрі **BSD**.

Однак всі програми з цього кола об'єднуються головною рисою – це доступність початкового тексту програми для громадського вивчення. Таким чином, частіше говорять про програмне забезпечення із відкритим вихідним текстом (англ. **open source software**), охоплюючи цим поняттям ширше коло програмного забезпечення.

З 1991 року багато чого змінилося в сфері інформаційних технологій: вільне ПЗ зростало, покращувалося, охоплюючи щоразу ширшу сферу. Особливо прогрес помітний кілька останніх років. Наприклад, донедавна спільноті бракувало якісного, швидкого переглядача Веб-сторінок – на основі відкритих вихідних текстів комерційної програми **Netscape** було створено **Mozilla Firefox**, який зараз стрімко захоплює ринок. Так само сталося і з офісним ПЗ – на основі відкриття вихідних текстів пакету **StarOffice** було створено **OpenOffice.org**, який за своїми можливостями практично не програє комерційним відповідникам. Графічні оболонки **KDE** та **GNOME** зараз пропонують досконалий, візуально привабливий і багатофункціональний графічний інтерфейс користувача. Саме ядро **Linux** теж зазнає постійних змін, спрямованих на покращення роботи пересічних користувачів настільних систем, наприклад, покращення аудіо-підсистеми досягнуло такого рівня, що виробники програмних синтезаторів почали використовувати **Linux** як вбудовану ОС для своїх виробів. Загалом прогрес у розвитку вільного програмного забезпечення засвідчує, що ряд представників такого ПЗ мають функціональні властивості, що не поступаються, а то й перевищують комерційні відповідники. Звичайно, залишаються і технічні проблеми, але на більшість із них можна сподіватися вирішення найближчим часом. Психологічні проблеми із освоєнням незвичного ПЗ можна побороти буквально за кілька тижнів. Тому ми рекомендуємо взятися за освоєння найпопулярніших представників вільного ПЗ.

## 1.2. Файлова система ОС **Linux**

Для довготривалого збереження інформації використовують диски (магнітні, оптичні і **USB**-диски [флеш-диски]). Запис інформації на диску називають файлом (англ. **file** – шеренга, ряд [байтів] – такий переклад відрізняється від загальноживаного, однак, на думку автора, краще передає зміст англomовного терміну). Кожен файл має своє ім'я, **Linux** дозволяє імена файлів довжиною до 256 символів. Для полегшення роботи з файлами рекомендують використовувати розширення імені файла, яке відділяють від імені крапкою. Рекомендовані розширення імен файлів:

**txt** – текстовий файл;

**htm, html** – текстовий файл з вбудованими посиланнями на інші джерела інформації, (широко використовують в Інтернеті);



xpm, jpg, gif, png – графічні файли;  
au, wav – звукові файли;  
z, tar, gz, tgz – архівні файли.

Розглянемо приклади імен файлів:

readme.txt

star.jpg

ruslana.wav

Файли можна об'єднувати у групи під якою-небудь назвою, наприклад **TEMP**, **MUSIC**, **WORK**. Такі поіменовані групи файлів називають каталогами (англ. *directory*, або *folder* – папка, тека). У каталозі крім вкладених у нього файлів, можуть бути інші каталоги. Всі каталоги, які розташовані на дисках ПЕОМ, об'єднані у дерево каталогів. На рис. 1 подана частина дерева каталогів файлової системи ОС **Linux**



Рис. 1. Дерево каталогів ОС **Linux**

Основний каталог диска (який є обов'язковим) називають кореневим і позначають знаком /. Кореневий каталог можна порівняти з стовбуром дерева, каталоги з гілками, а файли – з листочками дерева. Переміщення з одного каталогу до іншого дуже схоже до переміщення з однієї гілки дерева на іншу (якщо ви це коли-небудь робили або бачили).

Деякі каталоги створюються при встановленні (інсталяції) **Linux** і є стандартними. Користувач ПЕОМ з іменем *root* (суперкористувач) може змінювати структуру каталогів, однак це може призвести до втрати працездатності системи.

Розглянемо призначення частини стандартних каталогів:

- / - кореневий каталог;
- /bin - основні програми;
- /boot – у цьому каталозі розміщені файли, які необхідні для завантаження ОС;
- /dev – цей каталог містить файли, які відповідають пристроям ПЕОМ;
- /etc - файли конфігурації;
  - /passwd - інформація про користувачів;
  - /rc.d - сценарії ініціалізації системи;
- /home - домашній каталог (для користувачів);
- /lib - бібліотеки функцій та модулі ядра;
- /media - каталог для монтування змінних дисків (гнучких, оптичних і USB);
- /sbin - каталог для системних файлів;
- /tmp - для тимчасового зберігання файлів;
- /usr - програми і документи користувача;

Для повсякденної роботи на ПЕОМ бажано користуватись обмеженими правами доступу (див. далі), а усі створювані файли і каталоги розміщувати у каталозі **/home** (Домівка). Працюючи в ОС **Linux**, користувач може виконувати наступні дії із файлами і каталогами

<i>Дії із файлами</i>	<i>Дії із каталогами</i>
Створити – за допомогою програм опрацювання інформації ( і команди “ <b>Створити</b> ” контекстного меню, яка створить пустий текстовий файл)	Створити – командами операційної системи, командами контекстного меню або за допомогою спеціальних програм, які називають розпорядниками файлів (файловими менеджерами)
Скопіювати – командами операційної системи, командами контекстного меню або за допомогою спеціальних програм, які називають розпорядниками файлів (файловими менеджерами)	Скопіювати – командами операційної системи, командами контекстного меню або за допомогою спеціальних програм, які називають розпорядниками файлів (файловими менеджерами)
Перенести – командами операційної системи, командами контекстного меню або за допомогою спеціальних програм, які називають розпорядниками файлів (файловими менеджерами)	Перенести – командами операційної системи, командами контекстного меню або за допомогою спеціальних програм, які називають розпорядниками файлів (файловими менеджерами)
Видалити – командами операційної системи, командами контекстного меню або за допомогою спеціальних програм, які називають розпорядниками файлів (файловими менеджерами)	Видалити – командами операційної системи, командами контекстного меню або за допомогою спеціальних програм, які називають розпорядниками файлів (файловими менеджерами)
Перейменувати – командами операційної системи, командами контекстного меню або за допомогою спеціальних програм, які називають розпорядниками файлів (файловими менеджерами)	Перейменувати – командами операційної системи, командами контекстного меню або за допомогою спеціальних програм, які називають розпорядниками файлів (файловими менеджерами)
Запустити на виконання (лише для програмних файлів і сценаріїв [файлів, які складені із команд операційної системи]) – командами операційної системи, командами контекстного меню або за допомогою спеціальних програм, які називають розпорядниками файлів (файловими менеджерами)	Увійти (відкрити) і вийти (закрити) - командами операційної системи, командами контекстного меню або за допомогою спеціальних програм, які називають розпорядниками файлів (файловими менеджерами)

Перераховані дії користувач може виконувати лише із своїми файлами і каталогами, які розташовуються у каталозі **/home** (Домівка). Лише суперкористувач (**root**) може змінювати файлову структуру **Linux**, однак зміна файлової структури може призвести до втрати працездатності операційної системи, тому для повсякденної роботи на ПЕОМ бажано користуватись обмеженими правами доступу (див. далі).

### 1.3. Розпочинаємо працювати з **Linux**

ОС **Linux** розрахована на роботу з ПЕОМ багатьох користувачів. Для того, щоб забезпечити збережаність файлів користувачів, у цій системі використовують імена користувачів і паролі, які визначають права доступу користувача до ресурсів ПЕОМ (файлів, дисків, каталогів, програм). На ПЕОМ обов'язково є суперкористувач з іменем **root**, при встановленні **Linux** потрібно задати пароль для цього користувача. Пароль суперкористувача

варто не розголошувати і використовувати лише для налагодження системи. Для повсякденної роботи потрібно визначити імена користувачів і їх паролі та права доступу. Ці імена і паролі задає суперкористувач. При придбанні ПЕОМ з попередньо встановленою ОС **Linux** задайте пароль суперкористувача і пароль хоча-б одного користувача з обмеженими правами (повсякденна робота з правами суперкористувача небезпечна, оскільки ви випадково можете видалити важливі для операційної системи файли).

Після ввімкнення і завантаження необхідних системних програм на екрані дисплея з'явиться вікно реєстрації в системі:

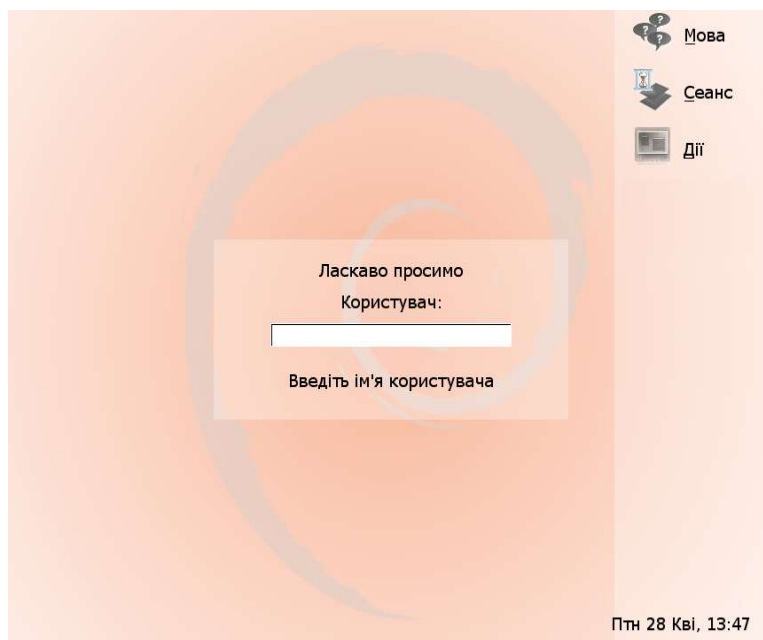


Рис.2. Реєстрація в ОС **Linux**

**Мова** – вибір мови сеансу, **Сеанс** – вибір графічної оболонки для сеансу, **Дії** – вибір одного із можливих варіантів дій (перезантажити, вимкнути і т.ін)

#### 1.4. Графічний інтерфейс в ОС **Linux**

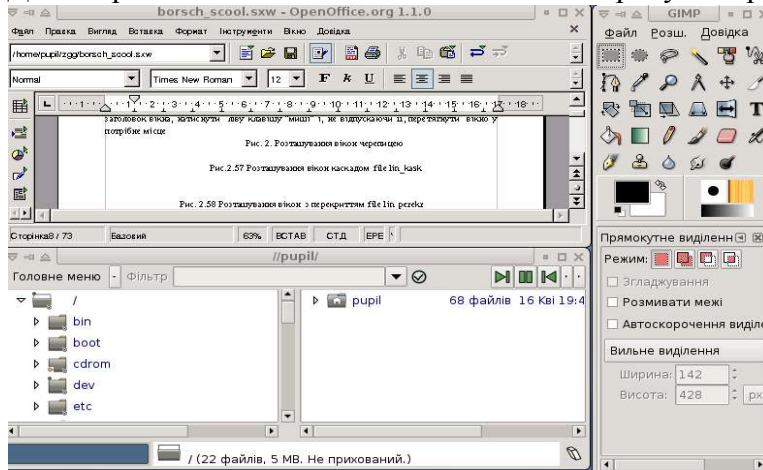
Графічний інтерфейс уперше був реалізований у 1982 році у дослідницькій лабораторії фірми **Xerox** для робочої станції **Xerox Star**. У 1985 році графічний інтерфейс з'явився у ПЕОМ **Macintosh II** фірми **Apple**. Ще пізніше графічний інтерфейс був реалізований фірмою **Microsoft** у графічних оболонках **Microsoft Windows x** (**Microsoft Windows 1.0**, **Microsoft Windows 2.0**, **Microsoft Windows 3.x**, **Microsoft Windows 95**, **Microsoft Windows 98**, **Microsoft Windows ME**) для IBM-подібних ПЕОМ. Графічна оболонка відслідковує переміщення координатно-вказівного пристрою (мишки, кулькового маніпулятора) на килимку і натискання клавіш цього маніпулятора і перетворює ці маніпуляції у команди операційної системи, які запускає на виконання. Для ОС **Linux** розроблено багато графічних оболонок, у цьому посібнику ми розглянемо лише три оболонки – **XFCE**, **GNOME**, **KDE**.

##### 1.4.1. Графічна оболонка **XFCE**

Різні графічні оболонки вимагають різних апаратних ресурсів (оперативної і відеопам'яті, швидкодії процесора, дискового простору і т.ін.). Оболонка **XFCE** у порівнянні з **GNOME** і **KDE** вимагає найменших апаратних ресурсів, тому при роботі на ПЕОМ з процесором **Celeron 466**, оперативною пам'яттю 128 Мбайт і вбудованим графічним

контролером вона матиме найшвидшу реакцію на дії користувача.

Графічна оболонка дозволяє відкрити багато вікон, кожне вікно відповідає окремій програмі, документу, диску або каталогу (течці). Вікна можна розташувати “черепицею”, каскадом або з перекриттям (рис. 3 - 5). Розміри вікон легко змінюються за допомогою “миші” шляхом “перетягування” зрізу (вертикального або горизонтального) вікна. Для цього наведіть вказівник "мишки" на зріз вікна (форма вказівника зміниться на стрілку з рискою), натисніть ліву клавішу “миші” і, не відпускаючи її, перетягніть зріз вікна у потрібне місце. Для переміщення активного вікна по екрану потрібно навести вказівник "мишки" на



заголовок вікна, натиснути ліву клавішу “миші” і, не відпускаючи її, перетягнути вікно у потрібне місце

Рис. 3. Розташування вікон черепицею

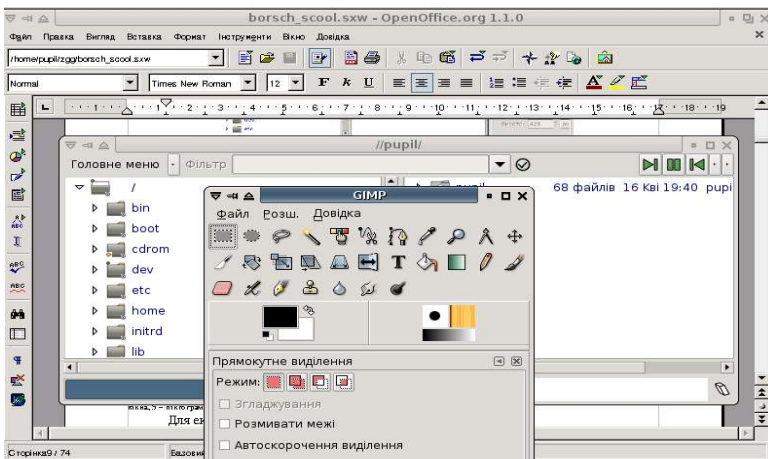


Рис. 4. Розташування вікон каскадом

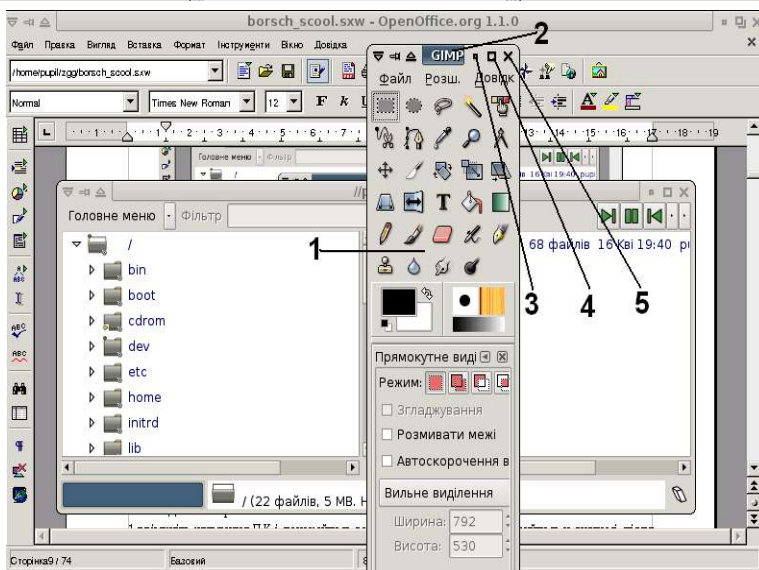


Рис. 5. Розташування вікон з перекриттям

1- активне вікно (заголовок вікна виділений темнішим кольором), 2- заголовок вікна (затискаючи ліву клавішу "мишки" на заголовку вікна, його можна пересувати у потрібне місце на екрані дисплея), 3 – піктограма прикриття вікна, 4- піктограма збільшення/зменшення розмірів вікна, 5 – піктограма закриття вікна

Для запуску програм в оболонці **XFCE** можна використовувати:  
 стартове меню;  
 запускачі програм;  
 контекстне меню **XFCE** (рис.7);  
 командний рядок.

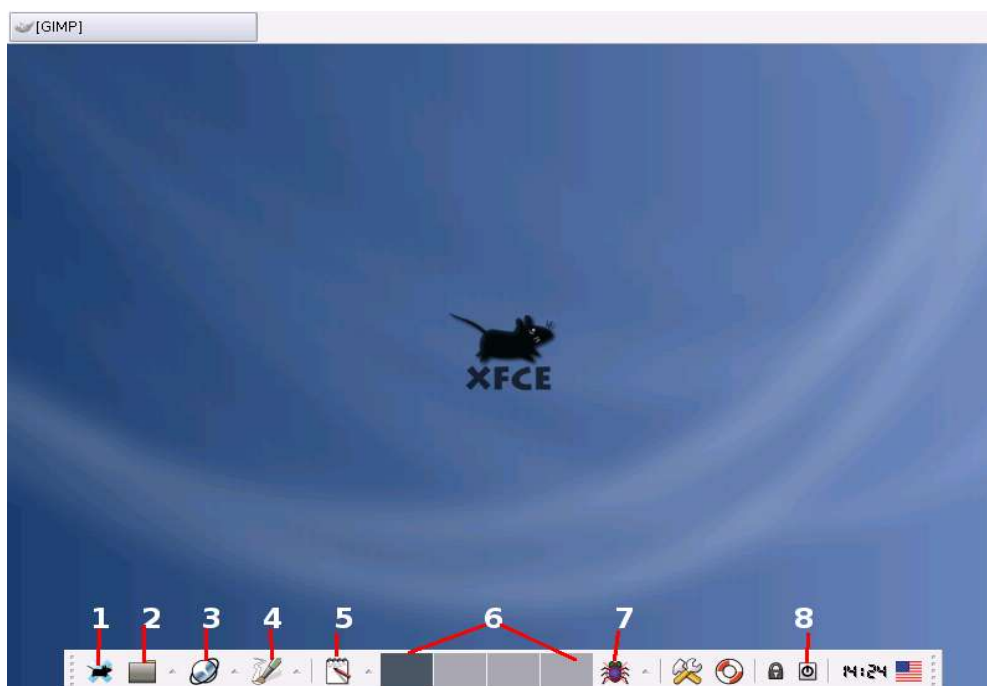


Рис. 6. Стільниця оболонки **XFCE**

1 - стартове меню, 2 – запускач розпорядника файлів **XFFM**, 3 – запускач програми перегляду Веб-сторінок, 4 – запускач **Gimp**, 5 – запускач **OpenOfficeWriter**, 6 - віртуальні стільниці, 7 – запускач програми **Algo**, 8 – завершення сансу роботи з **XFCE**



Рис. 7. Контекстне меню **XFCE** (контекстне і стартове меню в оболонці **XFCE** мають однаковий вигляд )

Для експериментів з вікнами виконайте наступну вправу:

1. увімкніть живлення ПЕОМ, дочекайтесь завантаження **Linux** і зареєструйтесь у системі, після цього на екрані з'явиться зображення “стільніці” (рис.6);
2. за допомогою запускатів (вказівники 2, 4, 5 на рис.6) запустіть розпорядник файлів **XFFM**, графічний редактор **Gimp** і текстовий процесор **OpenOffice.org.Writer**;
3. розташуйте вікна запущених програм черепицею. Для цього зменшіть розміри вікон програм так, щоб вони повністю помістились на екрані дисплею (для зміни розмірів вікна наведіть курсор мишки на зріз вікна [вертикальний або горизонтальний – форма курсора зміниться на стрілку з рискою], “затисніть лівою клавішею мишки” зріз вікна і переміщуйте його у потрібному вам напрямку. Для переміщення вікна по екрану дисплея “затисніть лівою клавішею мишки” заголовок вікна переміщуйте його у потрібному вам напрямку;
4. розташуйте вікна запущених програм каскадом;
5. розташуйте вікна запущених програм з перекриттям;
6. закрийте усі вікна (вказівник 5 на рис. 5).

У параграфі 1.2 перераховані дії, які можна виконувати із файлами і каталогами. Для виконання цих дій використовують спеціальні програми, які називають розпорядниками файлів (файловими менеджерами - англ. **File manager**). Кожна графічна оболонка має свій розпорядник файлів. В оболонці **XFCE** розпорядником файлів є програма **XFFM** (вказівник 2 на рис. 6). Після запуску на екрані **XFFM** дисплея з'явиться вікно

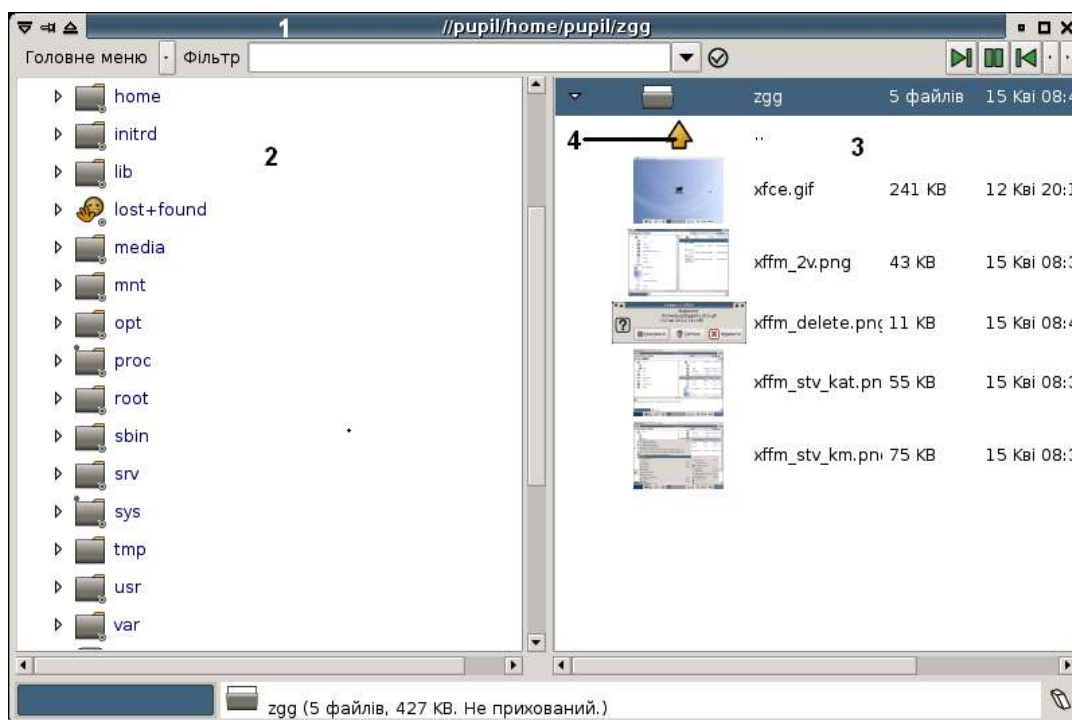


Рис.8. Розпорядник файлів **XFFM**

1 – заголовок вікна, 2 – ліве вікно, 3 – праве вікно, 4 – вихід із підкаталогу на один рівень вгору

Для виконання дій з файлами і каталогами можна використовувати контекстне меню



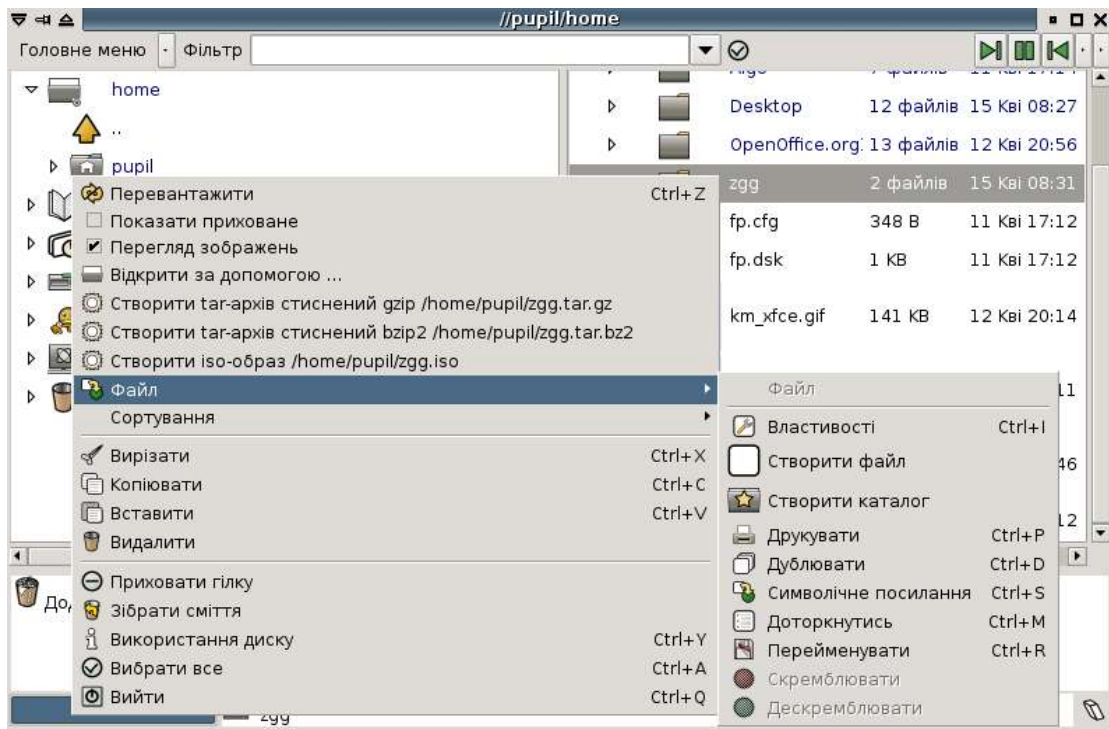


Рис.9. Контекстне меню XFFM

Для створення каталогу потрібно вибрати у контекстному меню “Файл – Створити каталог”. На рис.10. подано вікно із рядком задання імені нового каталогу (аналогічно створюють пустий файл)

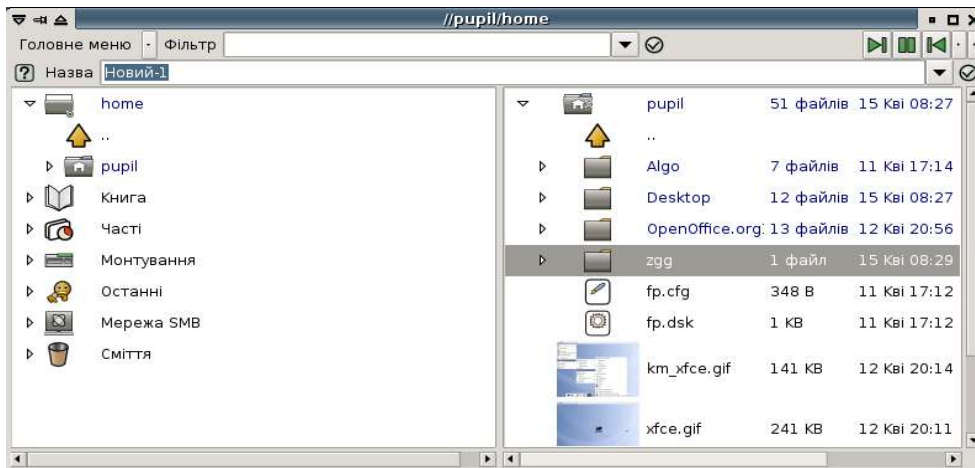


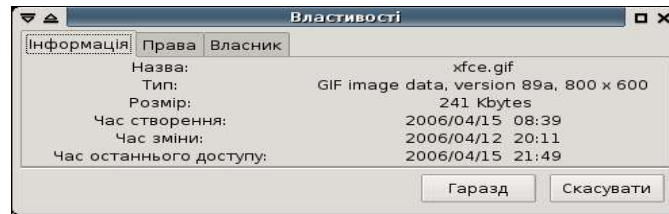
Рис.10. Створення нового каталогу

У рядку “Назва” замість тексту “Новий-1” наберіть ім'я створюваного каталогу.

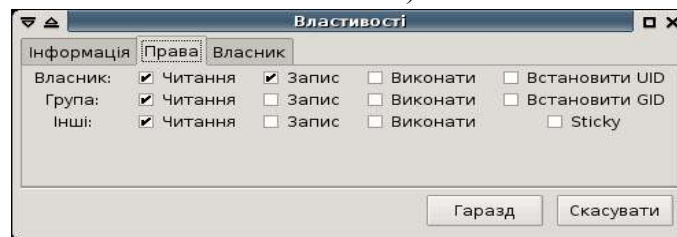
Для копіювання файлу або каталогу можна скористатись командами контекстного меню “Копіювати” у вікні із файлом, який потрібно скопіювати, “Вставити” у вікні із каталогом у який копіюється файл або каталог. Аналогічно виконується перенесення файлу або каталогу командами “Вирізати – Вставити”. Видалення файлу здійснюють за допомогою команди “Видалити”, перейменування - “Перейменувати”. Для роботи з гнучкими магнітними дисками потрібно увійти у каталог **media** (див. рис.8) а у ньому у каталог **floppy** – у відповідному вікні (лівому на рис. 8) XFFM відкриється список файлів і каталогів дискети. Після виконання операцій з дискетою її потрібно відмонтувати – виберіть лівою клавішею мишки каталог **floppy** і у контекстному меню (права клавіша мишки)

виберіть команду “Відключити”.

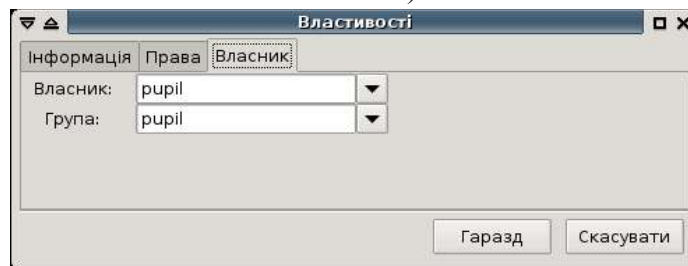
В ОС **Linux** файли мають своїх власників – користувачів, які ці файли створили. З метою збереження інформації власник файлу може обмежувати права доступу до своїх файлів. Для задання прав доступу до файлу можна скористатись командою “Властивості” контекстного меню. На рис. 11 подані вікна відображення властивостей файлу.



а)



б)



в)

Рис.11. Вікна відображення властивостей файлу

а- загальна інформація про файл, б – права доступу до файлу, в – інформація про власника файлу

У цьому прикладі власник файлу може зчитувати файл і записувати зміни у цей файл, члени його групи (якщо така група користувачів задана адміністратором ПЕОМ) і інші користувач можуть лише зчитувати файл. Власник файлу може змінити права доступу до свого файлу (“зняти мишкою” відмітки у відповідних віконцях). Для програмних файлів встановлення відмітки у віконці “Виконання” означає дозвіл на запуск цього файлу, зняття відмітки – заборону на виконання цієї програми певному користувачу. Таким чином адміністратор ПЕОМ може обмежувати права користувачів по запуску програм, наприклад, батьки можуть обмежувати доступ своїм діткам до глобальної мережі Інтернет. Текстовий вигляд прав доступу задано так: **rw-rw-r--**. Ці знаки поділяються на три групи:


перша група **rw-** означає права власника файлу ( **r** - файл можна читати, **w** - файл можна переписувати, - виконання для цього файлу не передбачено [ для програмних файлів третім символом у групі буде знак **x** - файл можна запускати на виконання]);

наступні три символи задають права доступу групи користувачів, до якої входить власник файлу;

останні три символи задають права доступу усіх інших користувачів системи.

Таким чином рядок **rw-rw-r--** означає, що файл, на який вказує вказівник "мишки", є звичайним (не програмним) файлом. Власник цього файлу і члени його групи можуть його читати і вносити у нього зміни. Усі інші користувачі системи можуть лише читати цей файл.



У цьому параграфі ми розглянули основні методи роботи з оболонкою XFCE. Залишилось лише в'яснити, як закінчити роботу з ОС Linux. Для завершення роботи потрібно "натиснути мишкою" піктограму із зображенням вимикача  у вікні реєстрації у системі вибрати пункт "Shutdown". Операційна система "завершить свої справи" (запише службову інформацію на ЖМД, закриє усі файли та відмонтує файлові системи) і вимкне блок живлення.

Якщо з яких-небудь причин ви не можете виконати завершення роботи Linux у графічному режимі, то виконайте наступні дії:

- перейдіть у текстову консоль натисканням клавіш [Ctrl]-[Alt]-[F1]
- натисніть клавіші [Ctrl]-[Alt]-[Del] для завершення роботи запущених процесів і операційної системи та перезавантаження ПЕОМ.

Якщо ви не хочете перезавантажувати ПЕОМ, то вимкніть блок живлення у момент виклику програми завантаження ОС.

### 1.4.2. Графічна оболонка GNOME

Після завантаження оболонки на екрані дисплея з'явиться зображення

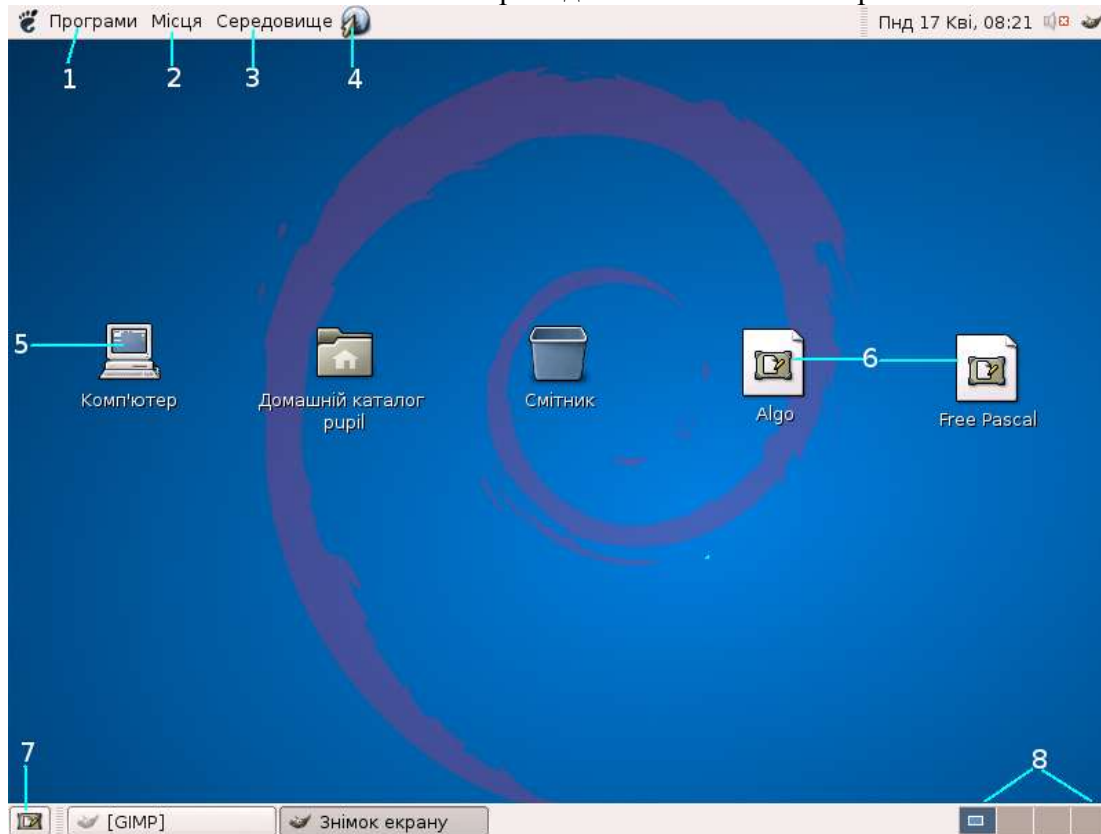


Рис. 12. Стільниця GNOME

1 - меню програм, 2 - меню роботи з файлами і каталогами, 3 - меню дій із GNOME, 4 - запуск програми перегляду Веб-сторінок, 5 - піктограма доступу до дискових пристроїв, файлової системи і локальної мережі, 6 - запускаті програм, 7 - прикриті усі вікна, 8 - віртуальні стільниці

Взаємодію користувача із оболонкою GNOME можна здійснити через меню стільниці, стільницю і панель стільниці. "Вказування мишкою" на об'єкти стільниці (піктограми і вказівники) оболонка перетворює у команди операційної системи, які вона запускає на виконання. Для запуску програм у GNOME можна використовувати меню програм і запускаті програм

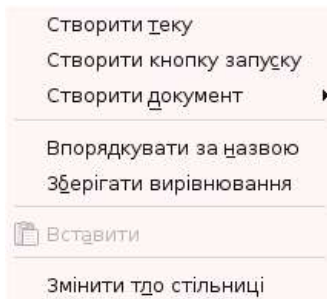


Рис. 13. Меню програм оболонки **GNOME**

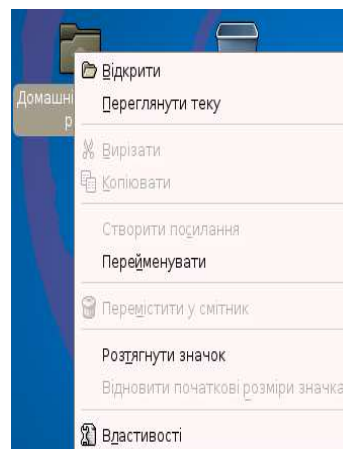
Для запуску програм із меню програм розкрийте його, “виберіть мишкою” потрібну групу програм, а у цій групі потрібну вам програму – на екрані дисплея розкриється вікно програми.

У меню “Місця” зібрано операції переходів по файлової системі і локальній мережі, у меню "Середовище" - дії із операційною системою.

Для роботи з об'єктами стільниці можна використовувати контекстне меню – меню, зміст якого залежить від вибраного “мишкою” об'єкта . На рис. 14 подані контекстні меню оболонки **Gnome** , які викликаються натисканням правої клавіші “мишки” на різних об'єктах стільниці



а)



б)

Рис. 14. Контекстні меню **GNOME**

а- контекстне меню при вказуванні на вільне місце стільниці і натисканні правої клавіші “мишки”, б - контекстне меню при вказуванні на каталог “Домівка” ( **Home**) і натисканні правої клавіші “мишки”

Для освоєння методів роботи з **GNOME** можна використати довідкову систему – “відкрийте **Програми – Довідка**”, виберіть потрібний розділ і освоюйте оболонку **GNOME** .

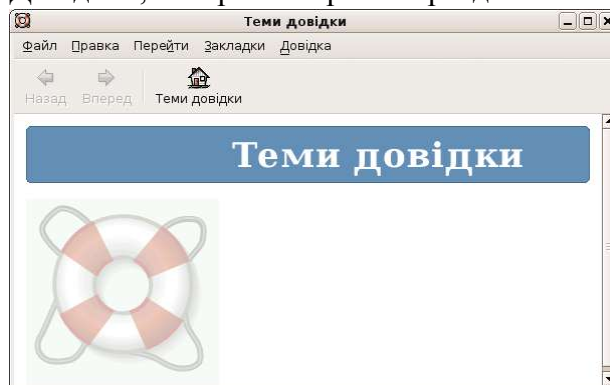


Рис. 15. Довідкова система **Gnome**

Для виконання дій із файлами і каталогами у **GNOME** використовують розпорядник файлів **Наутілус**

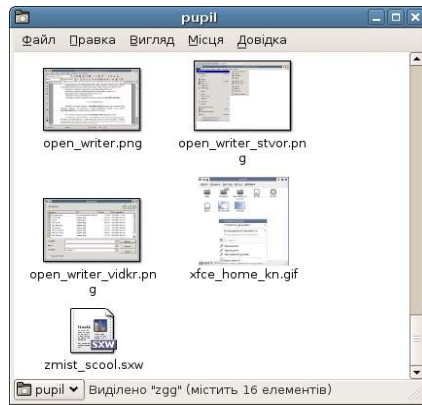


Рис. 16. Домашній каталог користувача pupii

Для створення нового каталогу або пустого текстового файлу можна скористатись контекстним меню **GNOME** – клацніть правою клавiшею мишки на пустому місці вікна програми **Наутілус** і виберіть **“Створити теку”** для створення нового каталогу або **“Створити документ – Порожній файл”** для створення текстового файлу.

Для копіювання, перенесення, перейменування і видалення файлу або каталогу можна використовувати контекстне меню **GNOME** або ж меню програми **Наутілус**

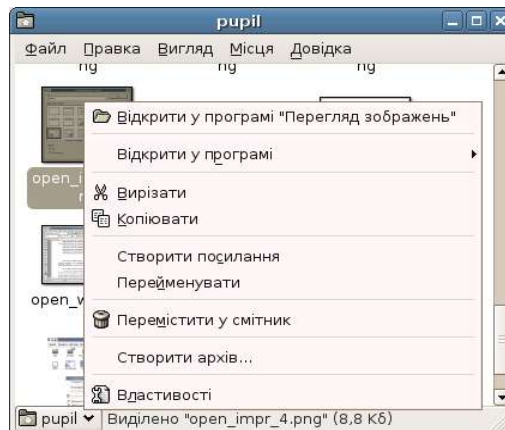


Рис. 17. Контекстне меню дій із файлом

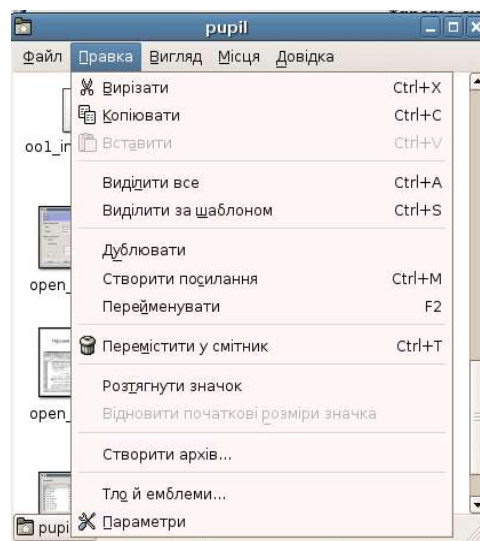


Рис. 18. Пункт "Правка" меню програми Наутілус

## Робота із дисками в оболонці **GNOME**

На сучасних ПЕОМ використовують такі дискові пристрої:  
жорсткі магнітні диски (на них розміщують файлову систему ОС **Linux**);  
гнучкі магнітні диски (**floppy**);  
оптичні диски (**CD-ROM, CD-RW, DVD, DVD-RW**);  
**USB-диски**.

Гнучкі магнітні диски, оптичні диски та **USB-диски** належать до змінних пристроїв (у ПЕОМ встановлено пристрій читання інформації з них, а самі диски ми встановлюємо за потреби). Сучасні збірки зазвичай налаштовані на автоматичне монтування змінних дисків – вставте оптичний диск у відповідний пристрій або під'єднайте **USB-диск** до системного блоку і на стільниці з'являться піктограма **USB-диска** і вікно програми **Наутілус** із його вмістом

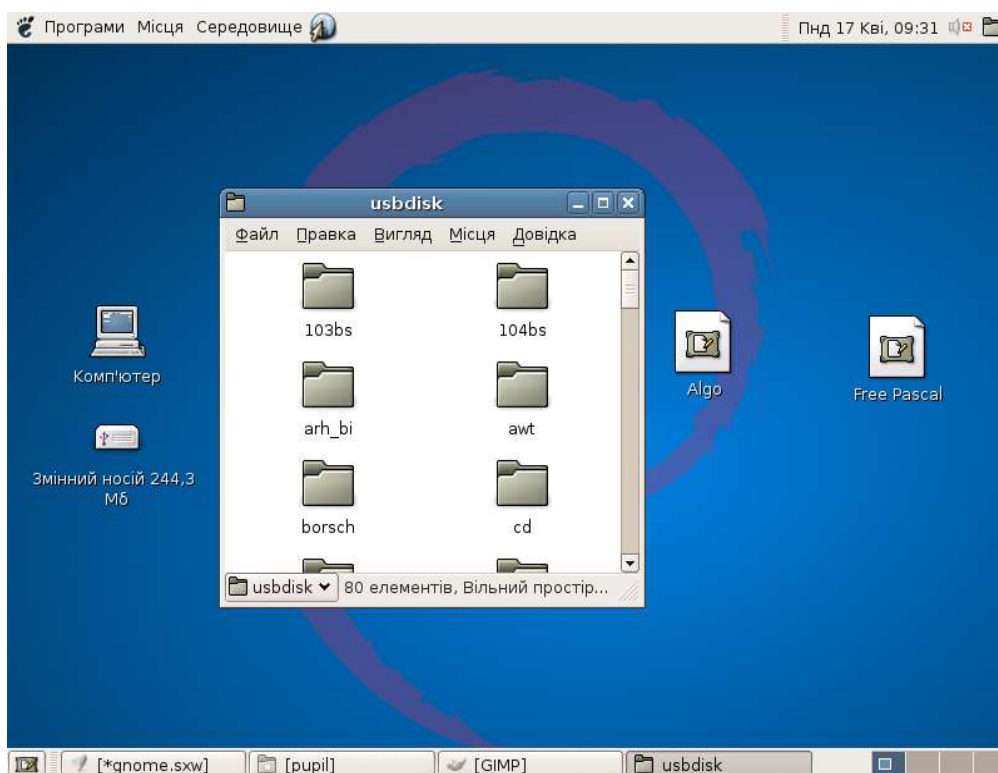


Рис. 19. Стільниця **GNOME** із змонтованим **USB-диском**

Файлові системи змінних дисків умонтовуються у каталог **media** дерева каталогів ОС **Linux**. Для того, щоб пересвідчитись у цьому, двічі “клацніть” лівою клавшею мишки по піктограмі “**Комп'ютер**” і у вікні, яке відкриється, по піктограмі “**Файлова система**” - у вікні розпорядника файлів “**Наутілус**” відкриється дерево каталогів ОС **Linux**. Відкрийте каталог **media** і ви побачите підкаталоги **cdrecorder** для оптичного диска, **floppy** для ГМД, **usbdisk** для **USB-диска**. Після завершення роботи із змінними дисками їх обов'язково потрібно відмонтувати, це можна зробити за допомогою контекстного меню (права клавша мишки) – відповідні піктограми зникнуть із стільниці.

### 1.4.3. Графічна оболонка KDE

Оболонка **KDE** є найбільш наповненою серед графічних оболонок для ОС **Linux**, але і потребує найбільших ресурсів ПЕОМ для своєї роботи. Після завантаження оболонки **KDE** на екрані дисплея з'явиться зображення стільниці **KDE**

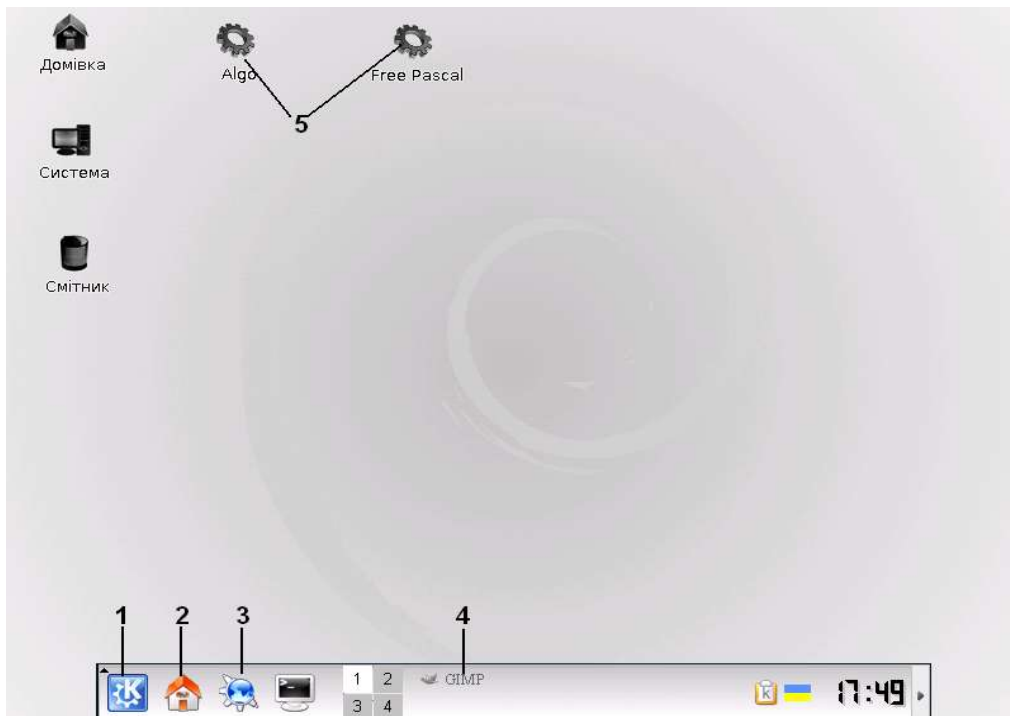


Рис.20. Стільниця **KDE**

1 – піктограма розкриття стартowego меню, 2 – перехід у домашній каталог, 3 - запускатч програми перегляду Веб-сторінок, 4 – згорнуте вікно програми **Gimp**, 5 - запуск програм **Free Pascal**, **Algo**

Для запуску програм можна використовувати:

- стартowe меню **KDE**;
- запускатчі програм;
- вказівники на програми;
- контекстне меню;
- командний рядок (**Виконати команду**).

Рис.21. Стартowe меню **KDE**



Для виконання дій з файлами і каталогами використовують розпорядник файлів



**Konqueror**, який можна запустити клацнувши по піктограмі .

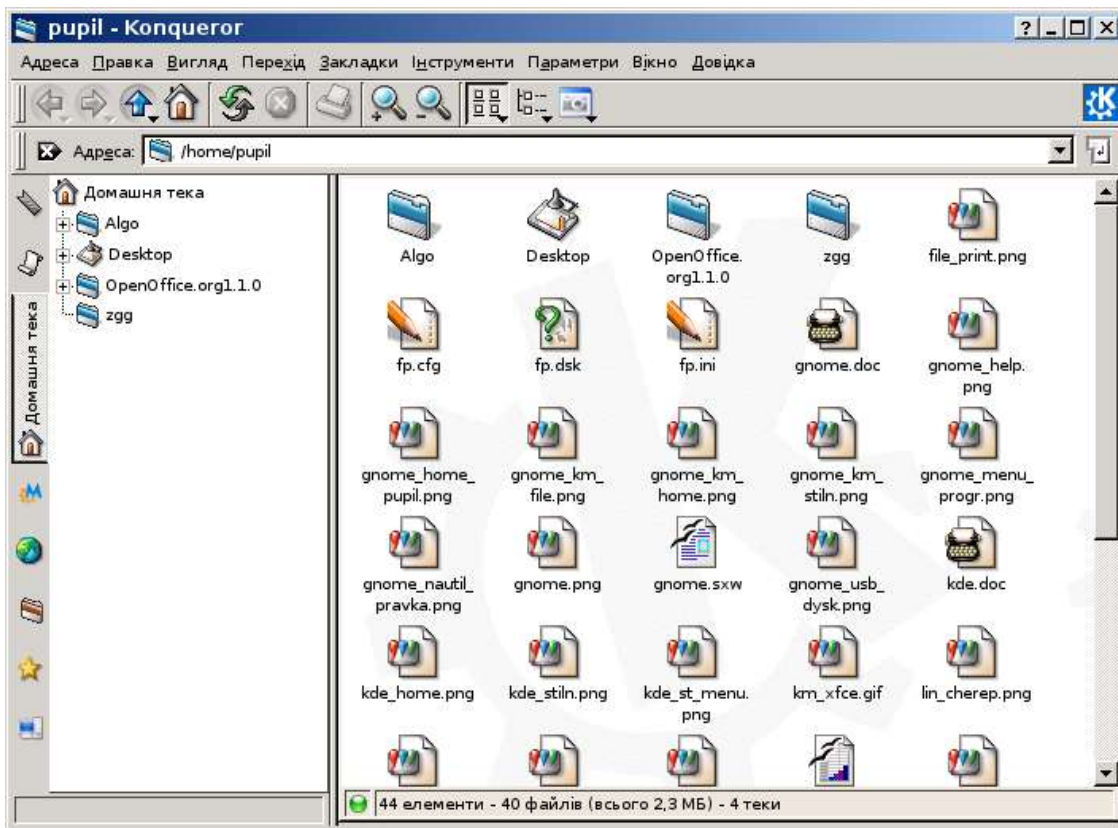


Рис.22. Розпорядник файлів **Konqueror**

Для створення каталогів і порожніх файлів можна скористатись контекстним меню KDE - “клацніть правою клавішею” мишки на порожньому місці вікна **Konqueror**.

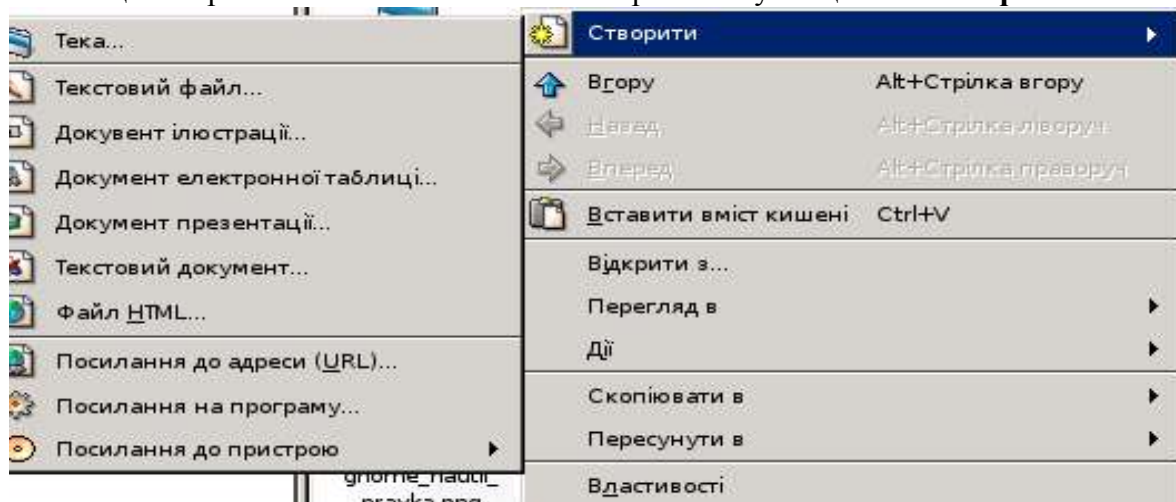


Рис.23. Контекстне меню **KDE**

Для копіювання файлів і каталогів можна використовувати багато методів, найпростіший із них полягає у “захопленні об’єкта лівою клавішею” мишки і перетягуванні його у потрібне місце. На рис. 24 подано вікно **Konqueror** з розділеним по горизонталі вікном (“**Вікно – Розділений перегляд Вгорі/знизу**”)

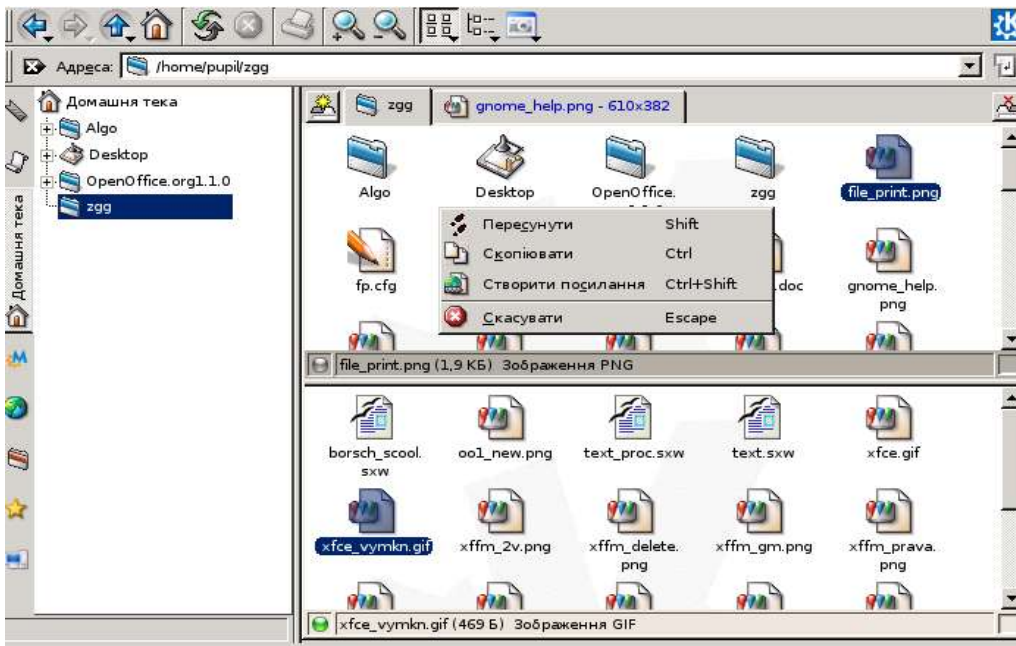


Рис.24. Контекстне меню для копіювання і переміщення файлів і каталогів

Програму **Konqueror** можна використовувати для переміщення по дереву каталогів **Linux** (рис.22) і перегляду Веб-сторінок (див. далі). Для переміщення по дереву каталогів вгору потрібно “клацнути лівою клавішею” мишки по інструменту “**Вгору**”. Для відкриття каталога (входження у каталог) – “клацнути лівою клавішею” мишки по піктограмі каталогу.

Для отримання довідки щодо роботи у **KDE** можна скористатись довідковою системою (пункт “Довідка” стартового меню **KDE** )

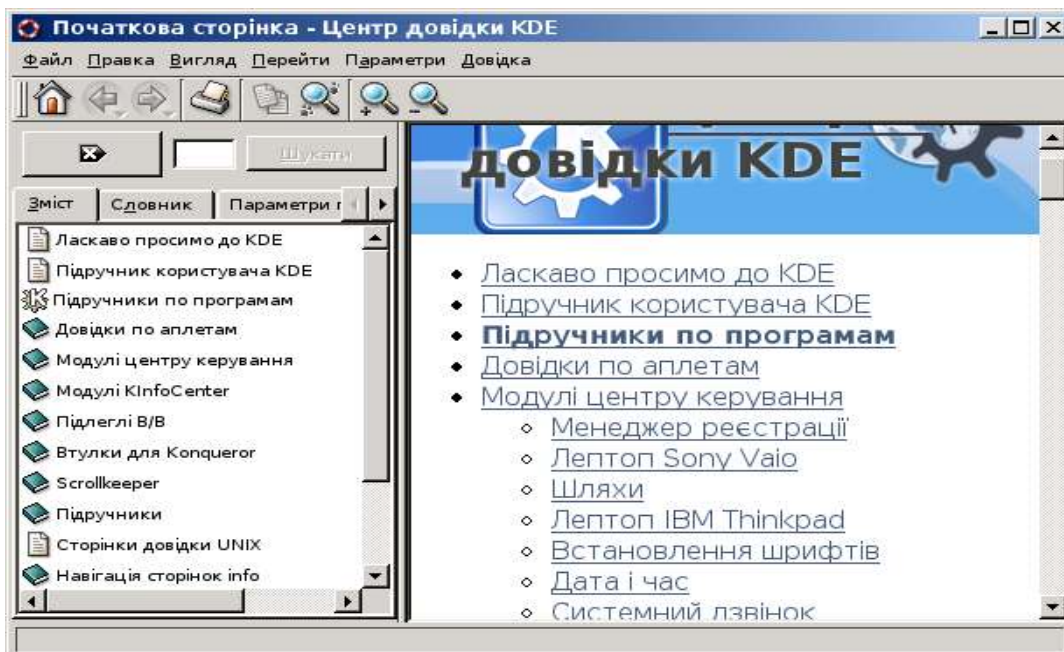


Рис.25. Довідкова система KDE

Користування **USB**-дисками у **KDE** дуже просте – вставте **USB**-диск у відповідне гніздо і через деякий час на екрані дисплея з'явиться вікно

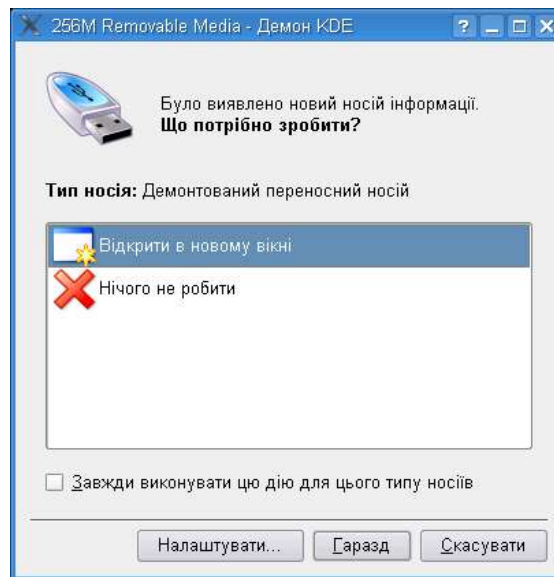


Рис.26. Пропозиції щодо можливих дій з **USB**-диском

Якщо ви погодитесь на “**Відкрити у новому вікні**” – на екрані дисплея відкриється вікно розпорядника файлів **Konqueror** з вмістом **USB**-диска. По завершенню роботи з **USB**-диском його потрібно відмонтувати. Для цього перемістіться по дереву файлів і каталогів на один рівень вище (інструмент **Konqueror** “**Вгорі**”) і відкрийте контекстне меню на піктограмі **USB**-диска

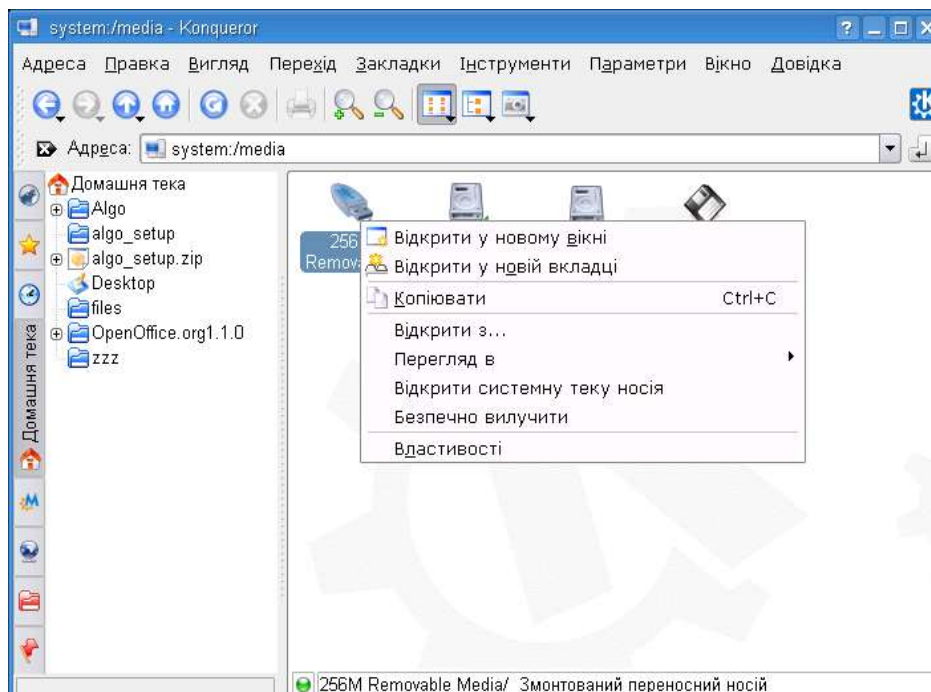


Рис.27. Демонтування **USB**-диска (**Безпечно вилучити**)



## 1.5. Командний рядок в ОС **Linux**

Історично склалося так, що для операційної системи **Linux** першочерговим є інтерфейс командного рядка – у відповідь на команду, набрану користувачем на клавіатурі, операційна система виконує певні зазначені дії над файлами, видає потрібну користувачу інформацію. Графічна оболонка у **Linux** вторинна, і слід зауважити, що часто “клацання” лівою або правою клавішею “мишки” на об'єктах стільниці графічна оболонка (**KDE**, **GNOME** або ін.) “перекладає” у набір команд. Попри існування графічних засобів для роботи **Linux**, найбільш універсальним і найпотужнішим інструментом залишається команда. Питання про самозрозумілість команд у порівнянні із графічною оболонкою досі відкрите, хоча експерименти показують, що для англомовних користувачів зрозумілість – не проблема, адже всі команди – це примітивізована англійська. У процесі виконання команд найважливішу роль відіграє спеціальна програма – **shell** (див. далі), командна оболонка, інтерпретатор команд. Вона забезпечує можливість обміну інформацією між програмами і виклик функцій ядра, інакше кажучи забезпечує інтерфейс доступу до функцій ядра. Командна оболонка потрібна для запуску будь-якої програми на виконання із вказаними параметрами, зміни пріоритету виконання і параметрів роботи, зміни режиму роботи програми, висилання їй сигналів на зупинку, припинення і т.д. Використовують її безпосередньо в текстовому режимі (режимі консолі) роботи ОС **Linux**, або в графічному середовищі (за допомогою програм-емуляторів терміналу). Існує кілька реалізацій цієї програми: **Bourne Shell (sh)**, **C sh (csh)**, **Korn Shell (ksh)**. Стандартним для **Linux** є варіант **bash (Bourne Again Shell)**. Окрім можливості запуску на виконання команд, що переважно містяться у зовнішніх файлах, кожен із командних інтерпретаторів має свій набір внутрішніх команд та команд управління, за допомогою яких можна скласти програму-сценарій (англ. **script**), яка буде виконувати послідовність дій. Власне вся структура **Linux** скріплена докупи за допомогою численних сценаріїв, котрі читають дані з конфігураційних файлів, виконують завдання і т.д. Таким чином, командний інтерпретатор – це також мова програмування, добре пристосована до виконання такого роду задач. Для роботи командної оболонки необхідно мати канал зв'язку до ядра (термінал). Ініціалізований термінал – це системна програма, що здійснює ввід з клавіатури (стандартний ввід) і передає вивід на дисплей (стандартний вивід). Після реєстрації у текстовому режимі відкривається термінал і перша програма, яка запускається для користувача – це командна оболонка. Після запуску командна оболонка ініціалізує необхідні змінні оточення і видає запрошення. Запрошення виглядає як короткий текстовий рядок з лівої сторони терміналу, що закінчується знаком долара **\$** для звичайного користувача і значком **#** для суперкористувача (системного адміністратора). Тому командну оболонку ще часто називають командним рядком, стрічкою. Зазвичай запрошення складається із імені користувача, назви машини і поточного каталога, у якому ви перебуваєте, хоч вигляд запрошення можна змінити. У командному рядку шляхом вводу команд можна маніпулювати файлами, управляти системою, запускати прикладні програми, наприклад розпорядник файлів **Midnight Commander (mc)**, аналог класичної програми **Norton Commander**. Робота у такому режимі нагадуватиме комусь роботу в **MS DOS**, хоча очевидно, можливості **Linux** у текстовому режимі значно переважають можливості **MS DOS**. У текстовому режимі у **Linux** можна здійснювати повноцінну діяльність – писати листи, тексти, переглядати Веб-сторінки, навіть слухати музику чи переглядати фільми, не просто хіба що редагувати графічні зображення. Для цього створено багато програм які можуть працювати у текстовому режимі, навіть можна створити багатовіконне середовище.

Отже, команди вводяться і запускаються на виконання за такою загальною схемою:

**запрошення#команда -ключ(і) <об'єкт дій команди> <спрямованість>**

Тобто у відповідь на запрошення вам потрібно набрати команду, ключі, вказати аргументи команди, після закінчення вводу натиснути клавішу вводу **Enter**. Об'єкт і спрямованість (аргументи) – необов'язкові компоненти, замість них команда може використати попередньо закладені в ній варіанти. Ключі команди дозволяють керувати результатами роботи команди, змінюючи спосіб дії команди від стандартного. Стандартно програми виводять результати та повідомлення про помилки на дисплей, і в разі необхідності вводу якихось даних очікує їх з клавіатури. Ввід-вивід можна переспрямовувати з(у) файл. Наприклад, команда видруку **ls (list)**

**...#ls**

видасть перелік всіх файлів у поточному каталозі (замість запрошення стоїть три крапки);

**...#ls -l**

видасть перелік всіх файлів у поточному каталозі з вказанням повних даних про атрибути файла (власника, дозволи, розмір), ключ **-l** походить від англійського **long** – *довгий*, довгий формат.

Змінимо спосіб виводу:

**...#ls -l > tt**

ця команда виведе інформацію вже не на дисплей, а у файл **tt**. У цьому прикладі **ls** – команда, **-l** – ключ, об'єктом дії є поточний каталог, який береться за замовчуванням, знак **>** позначає переспрямування виводу. Знак переспрямування вводу має вигляд **<**. Якщо виникає потреба спрямувати вивід однієї команди на вхід іншої, то використовується конвейер (англ. **pipe**), який позначається на допомогою символа **|**. Наприклад, команда

**...#ls -l | more**

виведе вміст поточного каталогу у довгому форматі на вхід до команди **more**, яка виводить результат поекранно (для перегляду наступного екрану натискають на клавішу прогалини). Така команда стає у нагоді, коли вміст каталогу не вміщається у один екран дисплею (стандартно 25 рядків), і надто швидко пробігає перед очима.

З практичної сторони управління системою зводиться до таких дій:

внесення змін в поточний стан системи (за допомогою команд);

внесення змін у конфігураційні файли (за допомогою спеціальних команд або безпосередньо виправляючи ці файли за допомогою будь-якого текстового редактора);

старт/перезапуск системних служб.

Слід наголосити, що командний режим не обов'язково означає однозадачність. Є ряд завдань, які для свого виконання не потребують взаємодії із користувачем, наприклад обробка масиву даних. Таку програму можна відправити на виконання на задній план, якщо в кінці команди додати значок **&**. Якщо команда виконується не так, як хотілося б, то її слід припинити, натиснувши комбінацію клавіш **Ctrl C**. Виконання команди можна призупинити, натиснувши комбінацію **Ctrl Z**. Спеціальна команда **fg** зможе потім повернути її до виконання. Для виходу із командного рядка є спеціальна команда **exit**, замість неї можна просто скористатися комбінацією **Ctrl D**.

Для прикладу розглянемо деякі найпоширені команди для роботи із файлами з поширеними ключами:

**ls list** – вивід переліку файлів і каталогів у поточному каталозі;

**ls -la** (видрук докладної інформації про файли, включно зі прихованими файлами у каталозі);

**ls -la file** (видрук докладної інформації про файл із назвою **file**);

**cd change directory** – перехід у інший каталог, зміна каталога;

**cd** (перехід до домівки, рідного каталогу поточного користувача);

**cd ..** (перехід у каталог вищого рівня) ;

**cd ім'я каталогу** (перехід у каталог з заданим іменем [нижчого рівня]);

**pwd print working directory** – вивести шлях до каталогу в якому ви зараз перебуваєте;

**mkdir ім'я make directory** – створити каталог з заданим іменем;

**rmdir ім'я remove directory** – видалити каталог із заданим іменем (порожній каталог);

**rm ім'я remove** – видалити файл з заданим іменем;

**rm -r ім'я\_каталога** – видалити каталог із всіма файлами і підкаталогами (застосуйте обережно!);

**cp ім'я1 ім'я2 copy** – копіювати файл з іменем ім'я1 у файл ім'я2;

**cp -dpr /шлях/ім'я1/ /шлях/ім'я2/** – копіювати перший каталог разом із всіма файлами і підкаталогами у другий каталог, зберігаючи всі атрибути файлів;

**mv /шлях/ім'я /шлях/ім'я2 move** – перемістити один файл з одного положення в інше під іншим ім'ям;

**touch ім'я touch** – “торкнутися”, тобто змінити час створення і модифікації файла на поточні. Використовують для створення пустого нового файла.

**\$mount /media/floppy** – приєднає файловою систему ГМД (пристрій **/dev/fd0**) до каталога **/media/floppy/**.

**\$umount /media/floppy** – синхронізує стан файлової системи на дискеті з файловою системою ОС **Linux**, після цього дискету можна вийняти із пристроя. Слід пам'ятати, що видалення дискети із зчитувача до демонтування може призвести до непередбачуваних наслідків для файлів на ній.

**\$mount /media/cdrom** – приєднає файловою систему типу **ISO9660** із пристроя **/dev/cdrom0** до каталогу **/media/cdrom**. Після цього можна копіювати із цього каталога файли на внутрішній ЖМД. Коли оптичний диск примонтований, то ви не зможете витягнути його із пристрою читання за допомогою кнопки на його панелі. Це можна зробити лише після команди **\$umount /media/cdrom**, або ще краще **\$eject**, яка сама виведе ятку із диском.

З таким набором команд можна виконувати всі основні дії над файлами і каталогами. І тут виникає логічне запитання, скільки команд з ключами треба запам'ятати? Для роботи з командами зовсім не потрібно запам'ятовувати їх формат - достатньо ввести команду **man ім'я команди** і на екрані терміналу ви отримаєте довідку по цій команді.

Для перегляду довідки про команди **Unix (Linux)** з графічної оболонки **KDE** можна використати програму **Konqueror** – введіть у рядку задання адреси команду **man:** і у вікні перегляду ви отримаєте перелік усіх розділів довідки **man**

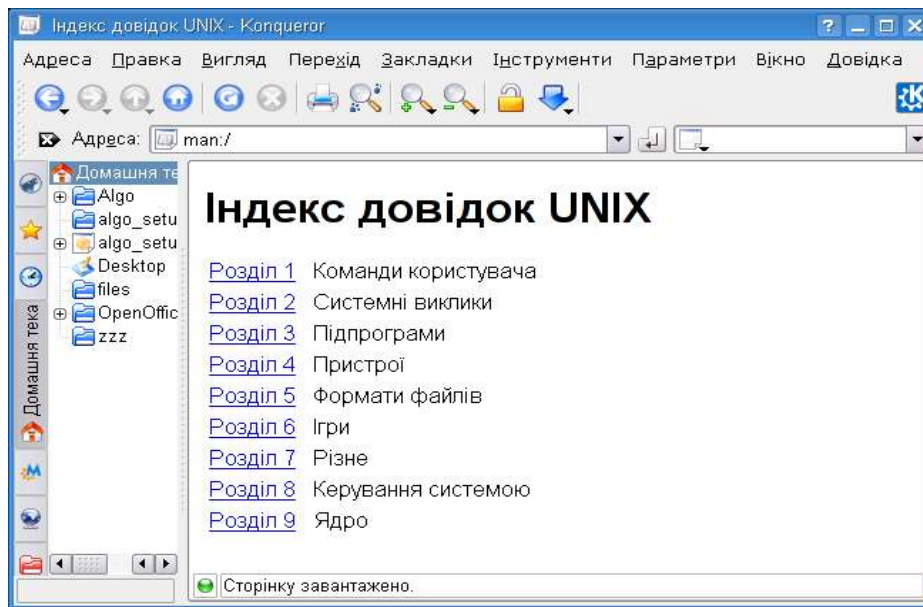


Рис. 28 Довідкова система **man**

Виберіть розділ довідки і команду – у вікні перегляду з'явиться довідка про вибрану команду. На зміну системі **man**, яка практикує вміщення всієї інформації в одну сторінку, деколи досить довгу, потрохи приходить інша система – довідкова система **info**. Для отримання довідки по команді **ls** за допомогою команди **info** потрібно відкрити термінал і ввести команду **info ls**.

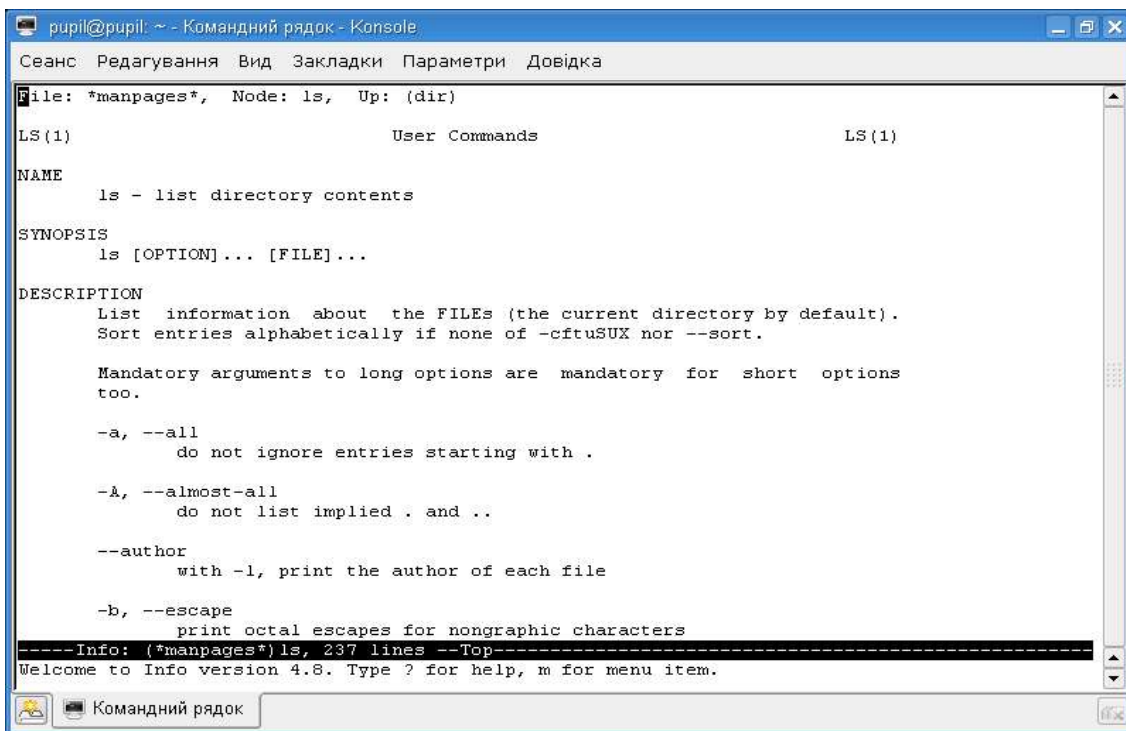


Рис.29. Довідкова система **Info** з довідкою про команду **ls**

Особливістю системи **Info** є те, що текст довідки структурований та пов'язаний посиланнями, що значно полегшує навігацію по ньому.

Важливим також є клас команд, які опрацьовують атрибути файлів. Розглянемо кілька прикладів:

**chmod** код ім'я *change mode* – змінювання прав доступу до файла (каталога) ім'я згідно із кодом;

**chmod u+x file.ext** – змінить атрибути файла так, щоб користувач-власник файла міг його виконувати. Встановлення біта виконання є обов'язковим для того, щоб файл став виконавчим, а виконання каталога полягає у можливості входження в нього. Замість **u** може стояти **g** для зміни прав членів групи, або **o** на позначення змін прав всіх інших користувачів. Знак **+** позначає надання певних прав, а знак **-** позбавлення їх. Права позначаються як звичайно – **r** (читання), **w** (запис), **x** (виконання);

**chown** *change owner*– зміна власника файла (каталога). Команда **chown user2.group2 file.ext** (передасть файл **file.ext** у власність користувача **user2** з групи **group2**);

Існує можливість локального перемикавання на обліковий запис іншого користувача, за допомогою команди **su** – ім'язапису *switch user*. У відповідь на запит слід ввести пароль користувача, на запис котрого ви перемикаєтесь. Всі ваші подальші дії будуть виконуватися від цього користувача, допоки ви не вийдете з його оболонки за допомогою команди **exit**. Переважно таким чином перемикаються у режим суперкористувача для виконання адміністративних задач.

На перший погляд ефективність команд неочевидна. Але головна потужність перелічених згори команд у тому, що вони можуть діяти не лише на один файл чи каталог, а на цілу множину, означену певним чином. Найпростіший випадок – застосування шаблонів назв файлів. Наприклад команда

```
...$rm /*~
```

усуне із поточного каталога всі файли, що закінчуються на знак **~**, це переважно запасні копії, що створюються текстовими редакторами. Можна будувати шаблони високої складності, використовуючи регулярні вирази.

На закінчення розглянемо доволі типовий приклад з практики, і заодно освоїмо ще кілька команд. Нам потрібно привести до порядку каталог із набором музичних файлів (у форматі **MP3**). Файли розкидані по різних підкаталогах, мають довгі назви, випадковим чином розподілених власників і різні права доступу до них. Отже спочатку слід перейти у цей каталог

```
...$cd /шлях/каталог
```

потім слід змінити власника всіх файлів і підкаталогів на користувача **user** з групи **group** у поточному каталозі.

```
...$chown user.group ./ -R
```

де ключ **-R** означає рекурсивний прохід по всіх підкаталогах донизу. Далі слід застосувати команду **chmod**, але так щоб вона по-різному діяла на каталоги і на файли. Для виділення переліку каталогів і переліку файлів окремо скористаємося командою **find** (знайти). Команда **find . -type d** дасть (знайде) перелік каталогів у поточному каталозі, а команда **find . -type f** дасть перелік файлів. Отже будемо конструювати

```
...$find . -type f -print0 | xargs -0 chmod o-w
```

Розшифруємо наші дії. Знаходимо всі файли із поточного каталога і підкаталогів, виводимо їх перелік, навіть якщо в їх імені є прогалина (для цього є ключ **print0**), передаємо через конвейер перелік на команду **xargs**, функція котрої полягає у переданні об'єкта команди на команду **chmod**, яка вже позбавляє всіх сторонніх користувачів прав не

виконання. Дуже лаконічно, чітко і швидко виконуються всі дії.

На цьому завершимо цей короткий огляд можливостей команд, і не торкатимемося великої теми, яка стосується найсильнішого застосування командного інтерпретатора – створення сценаріїв виконання. Цей інструмент дозволяє запрограмувати дії з управління системою, аналогічні до розглянутих нами, і таким чином виконувати цілу послідовність дій.

## 1.6. Адміністрування системи ОС **Linux**

Рано чи пізно, перед кожним користувачем ПЕОМ постають завдання керування операційною системою та встановленим у ній програмним забезпеченням, тобто кожному користувачу доводиться ставати на хвилину системним адміністратором. Ця необхідність може виникнути внаслідок несправності обладнання, потреби встановити нову версію якогось ПЗ, або надати доступ (чи його обмежити) до вашого комп'ютера для іншого користувача. Доволі часто доводиться переналаштовувати вже встановлені програми. Очевидно, у складних ситуаціях без послуг досвідченого адміністратора, озброєного спеціалізованими засобами, не обійтися. Але багато проблем може вирішити навіть звичайний користувач, лише слід зорієнтуватися в системі та володіти знаннями про внутрішню структуру ОС **Linux** та принципи її функціонування. Тому спочатку пропонуємо матеріал для ознайомлення, а далі висвітлюється кілька найбільш поширених операцій.

### 1.6.1. Зміст поняття ОС **Linux**

Уточнимо, що мають на увазі, коли говорять про “операційну систему **Linux**”. Цей термін вживається у двох значеннях. У первинному розумінні **Linux** – це центральна частина операційної системи (ядро, англ. **kernel**). Ядро і найближчі до нього супровідні програми – це власне те, що властиве саме **Linux**. Решта вільних програм можуть виконуватися і на інших платформах (навіть **Microsoft Windows**). Щоб спростити і автоматизувати процес встановлення і конфігурування, укладаються збірки **Linux** (дистрибутиви, англ. **distribution**). Це, наприклад, **RedHat Linux**, **Mandrake Linux**, **ASPLinux**, **Debian GNU/Linux**, **AltLinux** тощо. Вже у конкретних випадках збірки називають операційною системою **Linux** (у нашому випадку, операційна система **Debian GNU/Linux Etch**).

Збірка **Linux** складається з:

**програми встановлення**, за допомогою якої користувач в інтерактивному режимі (відповідаючи на питання та вибираючи із запропонованих варіантів) може встановити систему на ПЕОМ;

**самого наповнення** (системні і прикладні програми, системні і програмні бібліотеки і т.д.), у вигляді скомпільованих під цю архітектуру ЕОМ, готових до запуску виконавчих файлів, файлів конфігурації та додаткових файлів, складених у програмні пакунки;

**системи управління** програмними пакунками, що дозволяє здійснити розгортання наповнення збірки і подальший контроль над ним;

**вихідні тексти програм (джерела)**, на окремих додаткових дисках (щоб задовільнити умови ліцензії **GPL**, що вимагає поширення програм разом із вихідним текстом).

Така система реалізує ідею високої модуляризації наповнення збірки, простіше кажучи, збірка складається з багатьох окремих частин, пов'язаних між собою залежностями. Певні частини можуть бути змінені (видалені/поновлені) без потреби повного перевстановлення системи, і водночас підтримується система взаємних залежностей між цими частинами. В принципі, кожен може укласти свою збірку, за умови достатньої кваліфікації. Збірки можна звантажити із мережі Інтернет, купити копії лазерних дисків,

позичити або зробити ці копії самому. Можна також придбати коробковий варіант (з більшими видатками), що складається з дисків, кількох книжок, а також можливості технічної підтримки від укладача збірки протягом терміну від кількох місяців до року, коли на ваші запитання будуть відповідати фахівці.

### 1.6.2. Структура ОС Linux

Що ж таке операційна система? ОС можна уявити як посередника між ЕОМ і людиною. Саме вона дає змогу “оживити” ЕОМ, керуючи апаратною частиною, та подає людині зручні для керування інструменти. Системи типу **Unix** мають модульну будову, тобто ОС складаються з багатьох компонентів, котрі чітко виокремлюються і які можна представити шарами на умовній схемі.

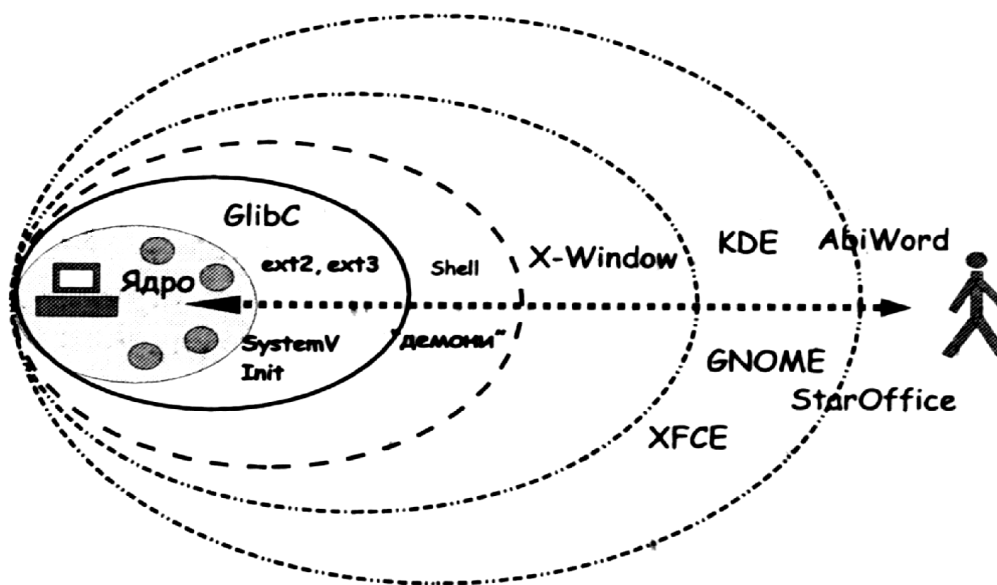


Рис.30. Структура ОС Linux

ПЕОМ може надати до використання такі ресурси як здатність до обчислень (час процесора), можливість для тимчасового і тривалого зберігання даних, засоби вводу-виводу. Ядро ОС **Linux** працює безпосередньо з цими ресурсами. Воно виконує такі основні завдання:

- управління обладнанням (підтримка “драйверів” пристроїв);
- управління процесами (програмами, що виконуються в системі);
- управління пам'яттю;
- здійснення операцій вводу-виводу;
- організацій взаємодії між процесами (пересилання сигналів);
- підтримка методів обробки сигналів з мережі (мережні протоколи);
- управління даними (файлова система).

З іншого боку, ядро – це величезна програма, написана мовою програмування **Cі** (близько 3 мільйонів операторів) і вже скомпільована. Ядро завантажується на самому початку запуску системи, і діагностичні повідомлення ядра проходять по екрану першими. Ядро доповнюється програмами супроводу та загальносистемними бібліотеками **GNU** для **Cі** (**glibc**), необхідними допоміжними програмами та системою ініціалізації **System V**-типу, укладеною файловою системою типу **ext2** чи **ext3**. Власне цей мінімальний набір і є тим, що називають операційною системою **Linux** у вузькому сенсі.

Ядро, як самостійна програма, має своє позначення версій і підверсій, виправлень, наприклад 2.6.8. Перша цифра позначає покоління, друга – версію. Ці дві цифри несуть основну інформацію, тому часто кажуть “**Linux 2.6**”. Третя цифра позначає підверсію. Підверсії ядра змінюються доволі швидко, під час внесення неглобальних змін або виправлення помилок. Ядро постійно перебуває у оперативній пам'яті ПЕОМ. Для того, щоб підтримати різноманітне обладнання, яке продукують для використання у ПЕОМ, і водночас не перевантажувати пам'ять зайвими даними, ядро **Linux** має модульну структуру. Тобто є основна частина ядра, яка містить базові функції (наприклад базові функції вводу-виводу), а підтримка конкретного обладнання винесена у модулі ядра. Модулі ядра перебувають на диску, їх можна завантажувати у пам'ять за потреби, та вивантажувати, коли їхні функції не потрібні. Між модулями ядра існують теж залежності, бо для роботи одного модуля часто потрібно довантажити ще один або кілька. У модулі виносяться переважно підтримка звукових карт, мережних адаптерів, додаткових накопичувачів інформації, інтерфейсів вводу/виводу. Комплектація ядра та те, які частини його будуть модуляризовані, є справою того, хто налаштував ядро до компіляції. Зазвичай це укладачі збірки **Linux**. У комплект збірки входять також вихідні тексти програми ядра. За їх допомогою можна заготовити потрібну конфігурацію для ядра та скомпілювати і встановити його замість стандартного. Таким чином, внутрішню будову системи **Linux** можна уявити як нашарування (оболонки) навколо центральної частини (ядра). Кожний шар призначений для виконання певних конкретних функцій у взаємодії з іншими частинами. Взаємодія всередині системи здійснюється аналогічно до взаємодії складових мережі, шляхом пересилання сигналів, система “прозора” стосовно мережі. Шари реалізують інтерфейс взаємодії з ядром. Частина програм працює в неінтерактивному режимі, в режимі сервера (у світі **UNIX** такі програми називають “демонами”). Вони очікують сигналів від програм-клієнтів, і відповідають на запити. Таким чином здійснюються всі службові функції сервера, наприклад, служби спільного доступу до файлів з мережі (служба **ftp**, сервер Веб-сторінок **httpd** з комплекту **Apache**).

Перший шар організації взаємодії користувача із системою – командна оболонка (англ. **shell**), інша назва – командний інтерпретатор, командний рядок. Він дозволяє інтерпретувати команди, отримані від користувача, висилаючи відповідні інструкції ядру через певні канали (терміналі). Наступна надбудова над командною оболонкою, що опирається на неї – це графічний інтерфейс користувача. У графічний інтерфейсу у першу чергу належить система **X Window**, яка здійснює взаємодію з відеообладнанням (відеокарта, дисплей) та засобами вводу (“мишка”, клавіатура), і працює в режимі сервера (**X-server**). До завдань **X Window** входить виконання найпростіших функцій – намалювати крапку певного кольору на дисплеї, пов'язати положення “мишки” на килимку з положенням курсора на дисплеї і т.д. Налаштування відеорежиму, роботи “мишки” і клавіатури здійснюється за допомогою налаштування системи **X Window**.

На базі функцій системи **X Window** будується кінцевий графічний інтерфейс користувача, наприклад **XFCE**, **GNOME** чи **KDE**. Такі графічні оточення опираються на графічні бібліотеки, у котрих реалізовано легкий доступ до складніших елементів графічного інтерфейсу: бігунків, панелей інструментів, меню, закладок і т.п. Графічне оточення складається із: менеджера вікон (закривання, відкривання, зміна розміру вікон); смужки задач; перемикача стільниць; розпорядника файлів; набору програм супроводу. Графічне оточення безпосередньо взаємодіє із користувачем.

### 1.6.3. Файлова система ОС **Linux**

Файлова система є однією із найбільш суттєвих складових операційної системи. Файлова система – це у першу чергу *спосіб організації* даних на носії, тобто спосіб



упорядкування і правила доступу до даних. Носієм може бути як локальний диск, так і диск сусідньої по мережі ЕОМ, навіть певний проміжок адрес оперативної пам'яті. Ідея файлової системи полягає у тому, щоб дати змогу прикладним програмам мати абстрактний інтерфейс до операцій вводу-виводу, не звертаючись кожній до особливостей роботи конкретного пристрою. ОС **Linux** підтримує велику кількість типів файлових систем (тобто може здійснювати операції читання/запису з відповідними розділами диску з такими файловими системами). Наприклад, якщо у вас паралельно встановлено на ПЕОМ ОС **Microsoft Windows**, то є можливість, перебуваючи у ОС **Linux**, зчитувати і записувати дані на розділи **Microsoft Windows**. "Рідними", а тому найбільш поширеними для **Linux** є файлові системи **ext2fs** та **ext3fs**. Сама організація файлових систем в **Unix** відрізняється від файлових систем ОС родини **Microsoft Windows 9X**, які базуються на таблиці розміщення файлів (англ. **file allocation tables, FAT**). Тут файлова система реалізовується за допомогою суперблока, котрий містить інформацію про файлову систему в цілому (або більшої її частини), індексний дескриптор (англ. **inode**) містить інформацію про номери блоків даних файле (або його частин). Назва файла разом із номерами індексних дескрипторів, що його описують, міститься у каталозі (англ. **directory**). Таким чином будується ієрархічна система, котра дозволяє знайти і зчитати відповідні дані, що належать певному файлові.

Логічну структуру файлової системи пов'язують із структурою носія за допомогою процедури форматування. У випадку ЖМД під час неї фізична одиниця носія – кластер (область однорідно намагніченого феромагнетика, яка зчитується за один раз) пов'язується із блоком даних. Під час розмови про ЖМД не уникнути пояснення поняття розділу або партиції (англ. **partition**). ЖМД має внутрішню так звану геометричну адресацію, котра задається положенням голівки читання/запису. За таким принципом диск можна поділити на розділи, інформація про поділ записується на початку диску у вигляді таблиці розділів (англ. **partition table**). Архітектура ПЕОМ дозволяє ОС **Linux** підтримувати до 4 головних розділів, або один із них замінити на розширений розділ, у якому можна утворити багато логічних розділів. Сучасні системні плати, як правило, дозволяють встановити до чотирьох ЖМД з інтерфейсом **Parallel** або **Serial ATA (IDE1, IDE2, IDE3, IDE4)**. Часом можна придбати системну плату із контролером **SCSI** (ЖМД з інтерфейсом **SCSI** є дорожчими від ЖМД з інтерфейсом **IDE**, їх переважно використовують у файл-серверах). Операційні системи **MS Windows xx (Free DOS)** використовують для позначення дискових розділів букви латинського алфавіту, починаючи з латинської **C:** (**C:**, **D:**, **E:**, і т.д.). ОС **Linux, Unix** використовують буквосполучення **hda, hdb, hdc, hdb** для ЖМД з інтерфейсом **IDE**; **sda, sdb, sdc, sdd** і т.д. для ЖМД з інтерфейсом **SCSI**. Розділи, які створені на диску **hda**, позначаються **hda1, hda2, hda3** і т.д. Відповідно **hdb1, hdb2, hdb3** є розділами диску **hdb**; **hdc1, hdc2, hdc3** - розділи диску **hdc**; **sda1, sda2, sda3** - розділи **SCSI**-диску **sda** і т.д. Розбиття ЖМД на розділи виконується менеджером дисків при встановленні ОС. При встановленні додаткового ЖМД розбити його на розділи можна командою **fdisk**. Для цього потрібно відкрити термінал і у командному рядку набрати:

```
fdisk /dev/hdb для диска hdb (Slave на IDE1);
```

```
fdisk /dev/hdc для диска hdc;
```

```
fdisk /dev/hdd для диска hdd.
```

На рис. 31 подано вікно терміналу з результатами роботи програми **fdisk /dev/hda** після введення команди **p** (вивід інформації про розбиття диска на розділи)

```

root@localhost:~ - Командний рядок - Konsole
Сеанс Редагування Вид Параметри Довідка
[root@localhost root]# fdisk /dev/hda

The number of cylinders for this disk is set to 4982.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
 1) software that runs at boot time (e.g., old versions of LILO)
 2) booting and partitioning software from other OSs
    (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): p

Disk /dev/hda: 255 heads, 63 sectors, 4982 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hda1  *           1          2550    20482843+   7  HPFS/NTFS
/dev/hda2                2551        2616       530145    5  Extended
/dev/hda3                2617        4982    19004895    83  Linux
/dev/hda5                2551        2616       530113+    82  Linux swap

Command (m for help): █

```

Рис. 31. Вивід інформації про розбиття диска **hda** на розділи (у цьому прикладі розділи **hda1** і **hda2** відведені для ОС **Microsoft Windows XP**, **hda3** – для ОС **Linux**, **hda5** – для області **Linux swap**)

Керування програмою **fdisk** здійснюють за допомогою команд (одна буква латинського алфавіту):

- a** – зробити розділ завантажувальним;
- d** – видалити розділ;
- l** – вивести список підтримуваних файлових систем;
- m** – вивести довідку про команди **fdisk**;
- n** – створення нового розділу;
- p** – вивід інформації про розбиття ЖМД на розділи;
- q** – вихід із **fdisk** без збереження змін;
- t** – зміна ідентифікатора файлової системи;
- u** – зміна одиниць вимірювання обсягу розділів;
- v** – перевірка таблиці розділів;
- w** – запис змін і вихід з програми;
- x** – додаткові функції.

В ОС **Linux** немає поняття логічних дисків **C: D:** і так далі, як прийнято позначати розділи в ОС родини **Microsoft Windows**. Вся система файлів складається в одне дерево із спільним початком, коренем (англ. **root**). Всі інші частини відгалужуються від цього початку, який позначають як **/**. Ці частини дерева не обов'язково мають перебувати на одному розділі чи диску, вони можуть перебувати навіть на різних ЕОМ, але під час ініціалізації системи вони поєднуються в одне ієрархічне дерево. Сам процес приєднання окремих дерев в точки вузлів називається монтуванням (англ. **mount**). Змінні носії (дискети, оптичні диски) також монтуються до загального дерева. Саме дерево системи файлів ОС **Linux** має стандартизовану будову, згідно із прийнятим виробниками збірок **Linux** погодженням. Кореневий каталог **/** має містити всі файли і каталоги, необхідні для запуску системи в режимі одного користувача: підкаталоги із виконавчими файлами основних утиліт **/bin** і системних програм **/sbin**, власними файлами суперкористувача **/root**, необхідними бібліотеками **/lib**, ядром для завантаження **/boot**. Також у кореновому каталозі розміщено загальний каталог для тимчасових файлів **/tmp**, каталог для примонтовування тимчасових носіїв **/media**. Тут також можна помітити підсистему **/proc** (ілюзорна файлова система, що

створюється у пам'яті ядром для представлення внутрішніх даних). Тут же міститься підсистема **/dev**, у якій розміщені файли пристроїв (в системах **Unix** доступ до пристроїв здійснюється аналогічно до доступу до файлів). Власне, основна маса самої операційної системи, що мало змінюється після встановлення, – файли прикладних програм, довідкова система, допоміжні файли, бібліотеки та засоби розробки, тощо — це все міститься під каталогом **/usr**, у вигляді розгалуженого дерева. Каталог **/var** призначений для швидкозмінних даних, як, наприклад, файли журналів, файли накопичення для пристроїв друку, електронної пошти, файли баз даних, файли, до котрих доступується через сервери, тощо. Всі конфігураційні файли, файли системи ініціалізації зосереджені у каталозі **/etc**. Позасистемні файли, тобто файли, які безпосередньо створюють і опрацьовують користувачі, дані користувачів цієї системи, зберігаються в особистих каталогах-домівках під каталогом **/home**. Звичайний користувач не має ані потреби, ані повноважень на змінювання файлів поза своїм власним каталогом (хіба тимчасових файлів). Таким чином, через помилку рядового користувача не постраждає ані система, ані файли інших користувачів. Всі системні файли мають своє визначене місце і відділені від файлів користувачів. Така структура файлової системи склалася історично і має глибокий практичний сенс.

Для роботи з файлами у текстовому режимі (у вікні терміналу) можна використовувати оболонку **Midnight Commander** (для запуску у командному рядку або відкритому терміналі наберіть команду **mc**). Оболонка **Midnight Commander** дуже схожа на оболонки **Norton Commander**, **Volcov Commander**, **Dos Navigator**, **Far**, **Windows Commander**. При роботі з **Midnight Commander** екран дисплею розділений на дві панелі

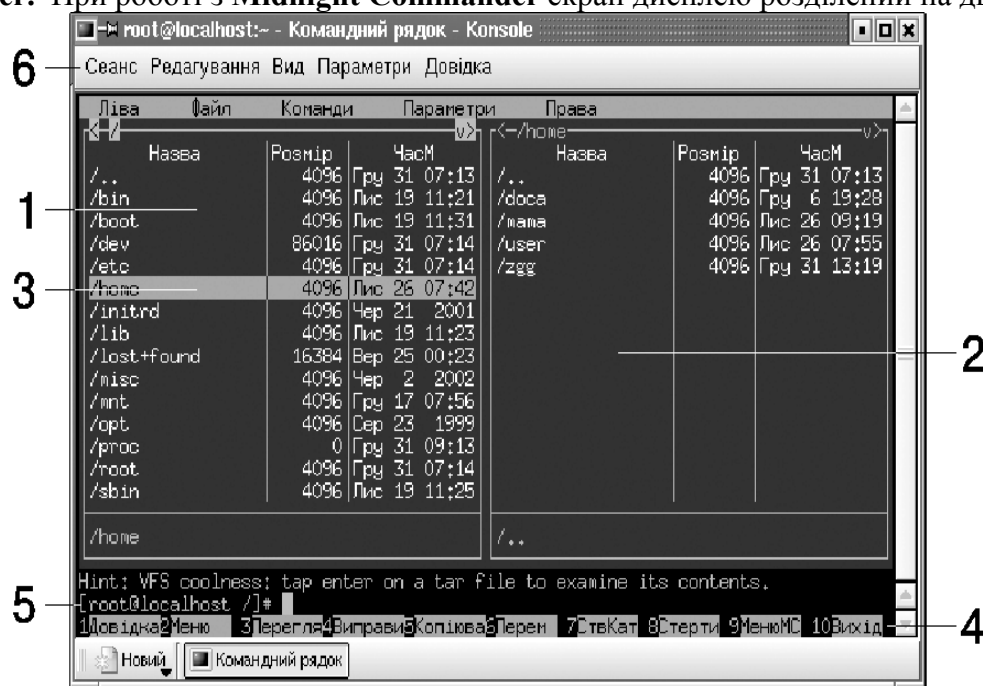


Рис. 32. Вікно терміналу з оболонкою **Midnight Commander**

1 – активна панель, 2 – пасивна панель, 3 – вибраний каталог (вибір файлу або каталога здійснюється переміщенням підсвіченого рядка по дереву файлів і каталогів клавішами керування курсором на клавіатурі або “вказуванням мишкою”), 4 – “гарячі” клавіші, 5 – командний рядок, 6 – меню **Midnight Commander**, F1 – виклик довідки, F2 – виклик меню користувача, F3 – перегляд текстового файла, F4 – редагування текстового файла, F5 – копіювання відмічених файлів і каталогів, F6 – переміщення або перейменування відмічених файлів і каталогів, F7 – створення нового каталога на активній панелі (на рис.3.61 подано вікно задання імені нового каталога), F8 – видалення відмічених файлів і каталогів, F9 – перехід у головне меню, F10 – вихід з **Midnight Commander**

Одна з панелей (на рис. 32 ліва) є активною, інша – пасивною. Більшість операцій виконують на активній панелі. Копіювання файлів (а також переміщення і створення символічного посилання) виконують з використанням обох панелей. На активній панелі

відображається вміст поточного каталога. Вибір активної панелі здійснюють “клацанням” лівої клавіші "мишки" або натисканням на клавішу [Tab] на клавіатурі. Переміщення по дереву файлів і каталогів здійснюють за допомогою клавіш керування курсором або “клацанням” лівої клавіші "мишки" у потрібному рядку. Для входу у каталог потрібно вивести підсвічений рядок на каталог і натиснути клавішу [Enter] (або двічі “клацнути” лівою клавішею "мишки"). Для виходу з каталогу потрібно вивести підсвічений рядок на знак /.. і натиснути [Enter] (або двічі “клацнути” лівою клавішею "мишки"). Для виконання дій з файлом (перегляд або редагування текстового файла, копіювання, переміщення, перейменування або видалення) його потрібно вибрати (вивести підсвічений рядок на файл або каталог) і натиснути відповідну функціональну клавішу. Окрім дій з окремими файлами або каталогами **Midnight Commander** може виконувати копіювання, переміщення, або видалення з групою файлів або каталогів. Для цього їх потрібно відмітити (натисканням клавіші [Insert]) і, після відмітки, натиснути відповідну функціональну клавішу (F5, F6, F8).

При видаленні непустих каталогів **Midnight Commander** зробить запит на підтвердження операції. Якщо у каталозі, куди ви копіюєте або переміщуєте файли, уже існує файл з таким же іменем, то вам потрібно буде уточнити, чи слід замінити старий файл. Для створення нового каталогу натисніть клавішу F7 (СтвКат) – на екрані дисплея відкриється вікно створення нового каталогу

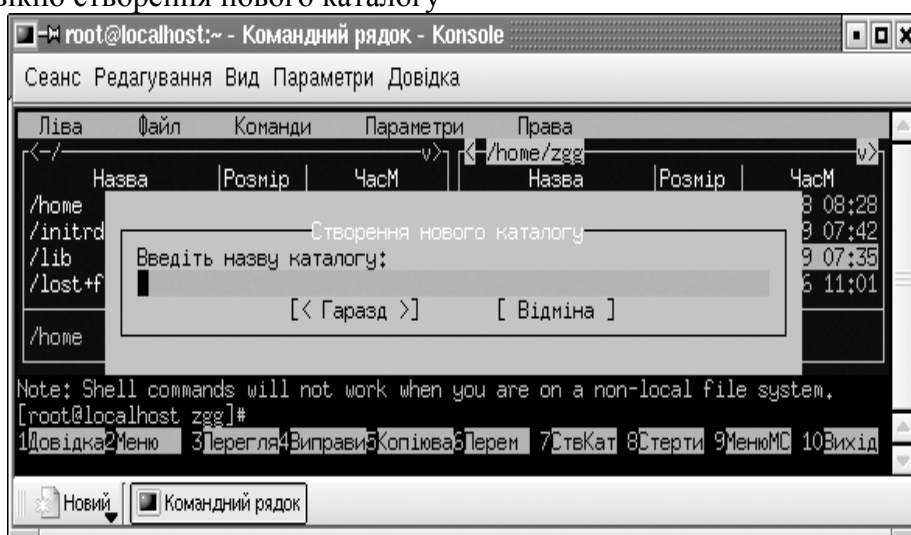


Рис.33. Вікно задання імені нового каталога

#### 1.6.4. Використання накопичувачів інформації

У спадок від **Unix Linux** також отримав і поняття “монтування” разом із командою **mount**. Саме слово походить від часів накопичувачів на магнітних стрічках, контролери котрих здійснювали команду монтування і лише після цього починали операції читання-запису. Головна ідея монтування – це “прикріплення” файлової системи певного носія до загального дерева. Місце (каталог), з якого починається файлова система носія, називається точкою монтування. Операція монтування здійснюється адміністратором командою вручну, або автоматично під час ініціалізації системи. Схема команди **mount**:

**...#mount -t <тип ФС> -o <опції> /dev/пристрій /точка/монтування**

Тип файлової системи – це, наприклад, **ext3 (Linux)**, **vfat (MS DOS, Microsoft Windows 9x)**, або **ntfs (Microsoft Windows NT, 2000, XP)**, або ж мережні файлові системи **smbfs**, **nfs** (у цьому випадку монтується носій взагалі із сусідньої системи через мережу), **udf** та **iso9660**

для оптичних дисків **DVD** та **CD-ROM**. Опції (модифікатори) йдуть через кому і відрізняються залежно від типу файлової системи. Для прикладу носії з файловою системою **vfat** можна монтувати із опціями **codepage=cp866**, **iocharset=koi8-u**. Завдяки цьому в **Linux** будуть коректно відображати назви файлів задані кирилицею в **Microsoft Windows 98/95**.

Найголовніший накопичувач інформації в ПЕОМ – це внутрішній ЖМД. Саме на ньому перебуває вся файлова система, у вигляді дерева з кореневим каталогом. Дисків може бути навіть декілька, і кожний із них ще може містити кілька логічних розділів (партицій [калька з **partition**]). Разом файлові системи різних дисків укладаються в деревоподібну систему. Здійснюється це є процесі ініціалізації системи, згідно із наповненням файлу **/etc/fstab**. У загальному випадку він виглядає таким чином:

```
# /etc/fstab: static file system information.
#
# <file system> <mount point> <type> <options> <dump> <pass>
proc /proc proc defaults 0 0
/dev/hda2 / ext3 defaults,errors=remount-ro 0 1
/dev/hda1 none swap sw 0 0
/dev/hdc1 /media/cdrom0 udf,iso9660 ro,user,noauto 0 0
/dev/fd0 /media/floppy0 vfat rw,user,noauto,sync 0 0
```

Із коментарів згори зрозуміло, яке призначення кожної колонки. Перша вказує на носій, друга – на точку монтування, третя – тип файлової системи, останні дві містять службову інформацію для програм резервного архівування. По рядках вказані: ілюзорна файлова система **proc**; точка монтування **/** розташована у другому розділі першого головного ЖМД (**hda2**); перший розділ цього диска (**hda1**) містить файлову систему підкачки (**swap**). Два нижні рядки містять записи для змінних носіїв. Змінні носії – це пристрої типу дискети (або ГМД), зчитувачів з оптичних носіїв, та різноманітних накопичувачів даних на основі під'єднання через **USB** (флеш-диски і переносні ЖМД). З точки зору операційної системи цифрові камери, які мають вбудовану флеш-пам'ять, теж є накопичувачами. Головна ідея таких пристроїв – носій можна змінювати на ходу, не вимикаючи системи. В **Linux** ці пристрої пов'язуються із спеціальними файлами в каталозі **/dev**, які у свою чергу монтуються до загального дерева файлової системи у каталозі **/media**.

Дискета – рудиментарний засіб перенесення інформації, але дотепер знаходить своє використання. До недоліків пристроїв ГМД слід зарахувати те, що вони не можуть передавати в ОС дані про свій стан. На дискетах використовують файлову систему **FAT**, запозичену із **MS DOS**. Щоб примонтувати дискету, слід виконати команду **\$mount /media/floppy**. Ця команда розшукає у файлі **/etc/fstab** відповідний запис щодо цього каталога, і приєднає файлову систему файлу пристрою **/dev/fd0** до каталога **/media/floppy0/**. Після цього можна маніпулювати файлами із нього за допомогою будь-якого розпорядника файлів або команд. Після закінчення роботи слід вийти з цього каталога та щоб жодна програма не тримала жодних відкритих файлів у цьому каталозі. Тоді команда **\$umount /media/floppy** успішно синхронізує стан файлової системи на дискеті, після чого носій можна видаляти із пристрою. Слід пам'ятати, що видалення дискети із зчитувача до демонтування може призвести до непередбачуваних наслідків для файлів на ній.

Поведінка оптичних носіїв дещо відрізняється. По-перше, ці носії працюють лише у режимі читання (запис на оптичні диски – справи спеціального ПЗ). По-друге, вони вже мають можливість контролювати стан носія. Команда **\$mount /media/cdrom** приєднає файлову систему типу **ISO9660** із пристрою **/dev/cdrom0** до каталогу **/media/cdrom**. Після цього можна копіювати із цього каталога файли на внутрішній ЖМД. Коли оптичний диск примонтований, то ви не зможете витягнути його із пристрою читання за допомогою кнопки на його панелі. Це можна зробити лише після команди **\$umount /media/cdrom** або ще краще **Seject**, яка сама виведе ятку із диском.

Ще інакше трактуються змінні носії на основі **USB**. Вони вважаються дисками типу

**SCSI**, точніше для них емулюється такий режим. Емуляція – це відтворення властивостей пристрою, якого реально не існує. Тобто, до під'єднання таких пристроїв жодних файлів в каталозі не існує, тим більше нема записів у файлі `/etc/fstab`. Коли такий пристрій вмикається у роз'язття **USB**, система **udev** оперативно створює файл пристрою, наприклад `/dev/sda1`. Далі поведінка системи буде відрізнятися, залежно від того, якою графічною оболонкою ви користуєтесь. **KDE** чи **GNOME** можуть вам допомогти, пропонуючи переглянути файли на носіїві, автоматично його змонтувавши. Інакше треба від імені суперкористувача виконати команду `#mount /dev/sda1 /mnt`. За потреби можна створити в системі додатковий запис у файлі `/etc/fstab`, наприклад: `/dev/sda1 /media/medium vfat rw,user,noauto,sync 0 0` та каталог `/media/medium`, і ваша флеш-карта, фотоапарат чи зовнішній накопичувач стануть доступними для монтування командою `mount /media/medium` і звичайному користувачу. Після закінчення роботи слід відмонтувати носій і видалити його з роз'язття.

### 1.6.5. Від увімкнення до роботи

Доброю ілюстрацією для розуміння структури системи і принципів її роботи може стати поетапне пояснення процесу ініціалізації. Після увімкнення ПЕОМ запускається вбудована програма **POST** (складова **BIOS**). Після успішного завершення програми самотестування ПЕОМ запускається програма завантаження операційної системи, яка здійснює пошук завантажувального запису на системних дисках (ГМД, ЖМД, зчитувач оптичних дисків) та у мережі. За інформацією, внесеною у завантажувальний запис, програма завантаження ОС знаходить ядро, завантажує і запускає його на виконання. У сучасних збірках **Linux** використовують переважно **GRUB** (великий уніфікований завантажувач, англ. **GRand Unified Bootloader**). Завантажувач дає змогу вибрати завантаження певної ОС, якщо їх кілька на одній ПЕОМ, або пропонує завантаження на різних ядрах **Linux**. Він встановлюється під час інсталяції самої системи і необхідний, навіть якщо на машині поставлено лише **Linux**. У випадку, коли завантажувач відсутній або пошкоджений під час встановлення іншої операційної системи, яка втручається на первинну доріжку ЖМД, то завантаження слід здійснювати з аварійної дискети. Завантажувач знаходить на диску скопільоване та архівоване ядро, розархівовує і завантажує його в оперативну пам'ять.

Спочатку завантажується монолітна (основна) частина ядра, яка містить найнеобхідніші драйвери пристроїв – наприклад драйвер контролера жорсткого диску, розпорядник оперативної пам'яті і пріоритетів процесів. Після завантаження монолітної частини ядра запускається первинний процес **init** з числовим утотожнювачем 1 (процес – це будь-яка програма, що виконується в ОС **Unix**, котрій виділено ресурси процесора), він породжує всі інші процеси. В багатозадачних системах існує поняття батьківського і дочірнього процесів. Кожен процес успадковує від батьківського область допустимої пам'яті, пріоритети на доступ до ресурсів, набір системних змінних. Первинний процес запускає системний сценарій ініціалізації і, далі, систему ініціалізації типу **System V**. Системний сценарій запускає найнеобхідніші служби, наприклад системний шриффт, встановлення дати і години, монтування (під'єднання) постійних розділів файлової системи, завантаження модулів ядра, тощо. Система ініціалізації типу **SystemV** складається із сценаріїв запуску окремих служб, які стартують (і зупиняються) у певному порядку. Наприклад, система типу **Debian** проходить через такі етапи (рівні виконання): рівень номер 1 (рівень без підтримки мережі, однокористувацький режим) та рівень номер 3 (багатокористувацький режим із підтримкою мережі). На цьому етапі практично всі служби запущені і працюють, можна реєструватися в системі у текстовому або графічному режимі. Між першим рівнем до третього завантажуються всі необхідні служби, які потрібні для роботи з ПЕОМ як робочою станцією чи сервером. Залежно від кількості покладених на ПЕОМ функцій, процес

завантаження системи може тривати доволі довго. Певний час може зайняти також апробація обладнання. Якщо встановити відповідну програму для графічної реєстрації, то система на третьому рівні видасть графічне запрошення на реєстрацію (англ. **login**). Після реєстрації користувача в системі запускається відповідне графічне оточення, наприклад **KDE**. Після виходу із графічного оточення система повертається до графічного запрошення на реєстрацію.

Тепер постає питання вимкнення. На графічному запрошенні до реєстрації є кнопка для вимкнення системи або її перевантаження. Після її застосування система пройде рівні ініціалізації у зворотньому порядку, вимкне всі служби і або самостійно вимкнеться (блок живлення з електронним вимикачем **ATX**), або ж видасть повідомлення про можливість вимкнення **Power down** (блок живлення з механічним вимикачем **AT**).

### 1.6.6. Управління користувачами

**Linux** – система багатокористувацька і пристосована для одночасного обслуговування діяльності великої кількості користувачів. Відповідно постає питання про розмежування цієї діяльності та забезпечення доступу до ресурсів ЕОМ кожному із користувачів відповідно до його повноважень. У всіх багатокористувацьких системах існує надзвичайно важливе поняття облікового запису (англ. **Account** – рахунок), як уособлення реального користувача-людини в системі. Облікового запис є сукупністю відомостей про реєстраційне ім'я користувача та пароль; повноваження і привілеї користувача; особистий каталог користувача (Домівка), заповнений особистими файлами і налаштуванням. Інформація про облікові записи користувачів зберігається у трьох системних файлах – **/etc/passwd**, **/etc/shadow**, **/etc/group**. Розглянемо формати цих файлів. Файл **/etc/passwd** складається із записів, кожен запис складається з семи полів, які розділяються двокрапками:

- реєстраційне ім'я користувача, яке має бути унікальним (в системі не повинно бути двох користувачів з однаковим іменем);
- зашифрований пароль (або \* у випадку використання системи тінювих паролів);
- числовий утотожнювач користувача (за цим утотожнювачем встановлюються власники файлів та процесів);
- числовий утотожнювач групи;
- інформація про користувача (переважно ім'я та прізвище);
- шлях до особистого каталога (домівки);
- командна оболонка, яку система запускає для користувача після реєстрації у системі.

Файл **/etc/shadow** використовують для створення системи тінювих паролів, у нього винесені з **/etc/passwd** всі паролі користувачів. Доступ до цього файлу надзвичайно обмежений. Таким чином, коли користувач реєструється у системі, то з файлів **passwd** і **shadow** добувається інформація про нього та звіряється пароль. Далі для системи користувач представляється через свій особовий числовий утотожнювач (англ. - **user identifier, UID**). Всі команди, які запускає користувач, пов'язані із його утотожнювачем. За ним визначаються пріоритети процесу у доступі до ресурсів ЕОМ. Користувачі можуть входити у групи, з тим щоб надавати певні привілеї сукупності користувачів. У файлі **/etc/group** задають зв'язки між іменами груп та числовими утотожнювачами груп (**GID**). Кожен запис у цьому файлі має чотири поля, розділені двокрапками – назва групи, поле для пароля, числовий утотожнювач групи, список реєстраційних імен користувачів, які входять у цю групу.

Файлова система ОС **Linux** побудована таким чином, що кожен файл чи каталог має свої атрибути (англ. **mode** – режим). За атрибутами файл належить певному власнику і певній групі, має відповідні дозволи на читання, запис, виконання для власника, членів групи і всіх інших користувачів. Інформацію про атрибути файла можна отримати за допомогою команди **ls** з ключем **-l**:

```
#ls -l file.ext
-rwxrw-r-- 1 user  user  43995 Гпу 5 2001 file.ext
```

Отримана інформація розбита по полях. У першому полі ліворуч віддруковано атрибути доступу. Перший символ **d** означатиме, що розглядається каталог, **l** – що розглядається символічне посилання, **-** – звичайний файл. Файлова система підтримує посилання на файл, тобто на певні частини носія інформації, поєднані у файлі, можуть посилатися з іншого місця файлової системи. Коли відбувається звертання до символічного посилання, то система перекидає його на те ім'я файла, на яке вказує символічне посилання. Можливе також створення жорсткого посилання, яке вказуватиме не на ім'я файла, а безпосередньо на індексні дескриптори, які його містять. Подальші літери у полі доступу позначають право на читання **r**, запис **w** та право на виконання **x**. Перших три символа подають права власника, наступні три – права члена групи, решта три – права всіх решти користувачів. Наявність знака **-** у певній позиції позначає відсутність відповідного повноваження. У другому полі виведено кількість жорстких посилань на файл (як мінімум одне). У третьому полі виведено реєстраційне ім'я користувача-власника, у четвертому – назва групи користувачів. Далі виведено величину файла у тисячах байтів. У шостому полі виведено час модифікації файла у вигляді дати і години. В останньому полі виведено назву файла.

Крім облікових записів, що пов'язані з конкретною особою користувача, в системі завжди присутні записи, які пов'язані із виконанням певних операцій для системного адміністрування. Такі облікові записи називають службовими. Таким псевдокористувачам належать системні файли, від їхнього імені виконуються команди і запускаються процеси, але ними не можна скористатися для входження в систему.

Особливе місце в багатокористувацьких системах посідає обліковий запис суперкористувача (привілейованого користувача, адміністратора) – **root**. Цей користувач в принципі не обмежений у своїх повноваженнях на здійснення змін у системі, а тому його пароль має зберігатися у суворій таємниці. Обліковий запис **root** має числовий утотожнювач 0, цей обліковий запис слід використовувати лише для адміністрування системи, виконання системних програм, і в жодному випадку не для запуску прикладних програм, чи виконання роботи звичайного користувача. Випадкова помилка у команді, введеній від суперкористувача, може коштувати системі “життя”. Пам'ятайте це! Змінювання прав доступу до файла (каталога) здійснюється за допомогою системної команди **chmod** (англ. **change mode**). Наприклад, команда

```
..#chmod u+x file.ext
```

змінить атрибути файла так, щоб користувач-власник файла міг його виконувати. Встановлення біта виконання є обов'язковим для того, щоб файл став виконавчим, а виконання каталога полягає у можливості входження в нього. Замість **u** може стояти **g** для зміни прав членів групи, або **o** на позначення змін прав всіх інших користувачів. Знак **+** позначає надання певних прав, а знак **-** позбавлення їх. Права позначаються як звичайно – **r**, **w**, **x** для читання, запису та виконання відповідно.

Зміна власника файла (каталога) здійснюється за допомогою системної команди **chown** (англ. **change owner**). Команда

```
...#chown user2.group2 file.ext
```

передасть файл **file.ext** у власність користувача **user2** з групи **group2**.



Існує можливість локального перемикавання на обліковий запис іншого користувача, за допомогою команди **su - ім'язапису**. У відповідь на запит слід ввести пароль користувача, на запис котрого ви перемикаєтесь. Всі ваші подальші дії будуть виконуватися від цього користувача, доки ви не вийдете з його оболонки за допомогою команди **exit**. Переважно таким чином перемикаються у режим суперкористувача для виконання адміністративних задач.

Введення у систему нових користувачів здійснює суперкористувач шляхом модифікації файлів, про котрі говорилося вище. Зручнішим за безпосереднє редагування може стати використання системних команд для утворення нового облікового запису, або програм із графічним інтерфейсом. Введення нового користувача у систему здійснюється командою

**...#adduser ім'я**

Ця команда внесе відповідні виправлення у конфігураційні файли, створить домашній каталог для користувача, і заповнить його стандартними файлами, скопійованими з каталога **/etc/skel**. Після введення користувача слід задати йому пароль для реєстрації. Це здійснюється за допомогою команди типу **passwd user2**, яка виконується від імені суперкористувача. У відповідь на запит слід двічі ввести пароль, і система присвоїть його користувачу **user2**. Для управління користувачами можна скористатися графічними утилітами, наприклад програмою **kuser**.

#### 1.6.7. Встановлення обладнання в ОС **Linux**

Сучасна ПЕОМ може бути оснащена широкою номенклатурою різноманітних пристроїв – від контролерів периферійних пристроїв до звукових пристроїв та відеокарт. Більшість із них супроводжуються програмною підтримкою виробника, який випускає для пристроїв драйвери для систем родини **Microsoft Windows** чи інших комерційних систем. Ситуація з підтримкою пристроїв в ОС **Linux** зовсім інша – бажано, щоб підтримка всіх можливих пристроїв вже була внесена у базовий набір збірки **Linux**. Встановлення драйверів із джерел, зовнішніх стосовно збірки – справа, що вимагає доброго розуміння процесу, і ми не будемо описувати таких випадків. Водночас налаштування обладнання, підтримка котрого внесена до збірки **Linux** є справою доволі тривіальною, якщо правильно дотримуватися процедури. Укладачі сучасних збірок досягли успіху у розробці програм встановлення збірок, і практично всі поширені пристрої ці програми правильно детектують автоматично, без вашої участі. Тому під час встановлення програма або підкаже вам правильний вибір, або взагалі не ставитиме питань.

Збірка **Debian GNU/Linux** нових версій контролює обладнання за допомогою системи **udev**. Вона в автоматичному режимі під час запуску здійснює апробацію апаратного забезпечення та створює від файли пристроїв в каталозі **/dev**. Це стосується базових пристроїв, наприклад, запускається підтримка не самого пристрою друку, а паралельного порта, до якого він увімкнутий.

Окремим і дуже важливим питанням є налаштування графічної системи **X Window**. В **Debian Linux** останніх версій ця функція реалізована за допомогою системи програм **X.Org**. Ця система містить підтримку практично усього спектра відеообладнання, у вигляді бази даних дисплеїв та набору власних модулів для підтримки графічних контролерів. Під час встановлення ОС **Linux** відбувається налаштування цієї підсистеми. Її можна переналаштувати за допомогою команди

**# dpkg-reconfigure xserver-xorg**

Після запуску цієї команди система конфігурування задасть вам ряд запитань про виробника графічного контролера (якщо не вдасться її впізнати автоматично), координатно-вказівний пристрій та клавіатуру, а також максимально можливу і потрібну вам роздільну здатність дисплея. Власне, результат конфігурування системи **X Window** – це правильно заповнений файл налаштування **/etc/X11/x.org**.

Для налаштування звукового обладнання вам слід установити пакунок **alsa-utils** разом із залежностями. Вже рядовим користувачем, в графічній оболонці, слід викликати програму мікшера і налаштувати рівень сигналу в каналах.

### 1.6.8. Методи встановлення програм в ОС **Linux**

Важливою складовою частиною збірки **Linux** є система управління програмними пакунками. Вона оперує із програмним забезпеченням у формі програмних пакунків. Програмний пакунок (пакет, англ. **package**) – це укладений архів, який містить у собі файли, потрібні для функціонування певної програми (або групи програм) разом з інформацією про те, де у системі вони мають бути розміщені, а також інструкціями автоматичного післяустановчого налаштування. Програмний пакунок спочатку має форму файла, а після встановлення перетворюється на набір файлів, розсіяних у відповідних місцях по системі, та записів у відповідній базі даних. Таким чином, у системі після встановлення кожен файл належить певному пакунку (тимчасових файлів і файлів користувачів це не стосується), і підтримується база даних про те, куди і що було встановлене. Поширюються програмні пакунки для нашої системи у вигляді файлів з розширеннями типу **.i386.deb**. Перша частина розширення позначає архітектуру процесора (**Intel 386**), під яку скомпільовані двійкові файли, або відсутність налаштування під архітектуру, як у другому випадку. Друга частина розширення власне позначає, що це програмний пакунок для збірки **Debian**. Цей варіант **Linux** має свою систему управління пакунками, що складається із низового рівня **dpkg**, та верхнього рівня **Apt**. Разом ця система дає можливість управляти, а саме поновлювати, встановлювати і забирати із системи будь-які компоненти. Слід пам'ятати – всі наведені команди виконуються від імені суперкористувача (перегляньте також підрозділ про управління користувачами для орієнтування), набираються у командному рядку після символу **#**. Крім командного інтерфейсу, існують програми з графічним інтерфейсом для виконання аналогічних завдань, наприклад **synaptic**. Після пояснення сенсу команд, графічний інтерфейс будь-якої програми управління пакунками буде легко зрозумілим. Всі команди управління програмними пакунками можна поділити за метою на команди інформаційні (запитів), перевірки, та команди модернізації, встановлення та видалення. Управління програмними пакунками зводиться або до роботи з базою даних вже встановлених пакетів, або до роботи з файлами пакунків.

Інформаційні команди (запиту) призначені для отримання інформації про вміст пакунків, поточний стан встановлених пакунків, належність певних файлів до пакунків і т.д. Команда

```
...#dpkg -l
```

виведе перелік всіх встановлених у системі програмних пакунків. Кожен пункт переліку міститиме інформацію про назву пакунку та його версію. Для прикладу наведемо один рядок.

```
ii zip          2.31-3      Archiver for .zip files
```

Тут перше слово подає назву програмного пакунку, далі – версія пакунку у вигляді трьох

цифр. Перша цифра позначає версію, друга – підверсію, а третя – варіант (виправлення). Після версії через дефіс виводиться версія перепакування, варіант пакунку (той самий пакунок може переукомплектовуватися кілька разів). Нумерування версій таким чином є традиційним для світу **Linux**. А тепер ми наведемо набір команд, що можуть стати корисними у практичному використанні.

Команда **...# dpkg -l | more** аналогічна до попередньої, але вивід інформації про встановлені пакунки буде відбуватися поекранно, якщо натискати клавішу прогаліни, або порядково, коли натискати клавішу **Enter**. Для припинення виводу треба натиснути **q**. Для конструювання цієї команди було двічі використано конвейер для переспрямування виводу.

Команда **...#dpkg -l > listpackages** так само виводить упорядкований за алфавітом перелік всіх встановлених пакунків, але тепер вже не на екран, а у файл **listpackages**. У цій команді було використано конвейер і переспрямування виводу. Файл **listpackages** можна пізніше надрукувати, докладно вивчати, порівнювати з переліком пакунків, встановлених на іншій ПЕОМ.

У випадку, коли увесь перелік не потрібний, але необхідно отримати інформацію про наявність певного пакунку, то стане у нагоді команда **...#dpkg -l | grep xxx**, яка видає перелік всіх пакунків, у назвах яких міститься вираз **xxx**. Така команда може бути корисною, якщо ви не пам'ятаєте чітко назву пакунку.

Для більш докладного вивчення вмісту вже встановленого пакунку, можна скористатися командою, яка виводить перелік всіх файлів, котрі належать певному пакунку:

**...#dpkg -L назва\_пакунку | more**

Можна розв'язати і обернену задачу. Для того, щоб з'ясувати, якому саме пакунку належить конкретний файл у системі, слід виконати таку команду

**...#dpkg -S /шлях\_до\_файла/файл**

Для встановлення, поновлення і видалення пакунків зручно використовувати систему **APT**. Вона дає змогу автоматично визначити потрібні за залежностями пакунки, стягнути їх з мережі та встановити, здійснивши після цього їх базове налаштування. Застосування цих команд дає змогу використати головну функцію системи управління пакетами – операції модифікації вмісту збірки. Слід зауважити, що перевстановлення цілої системи з метою доставити кілька програмних пакунків – далеко не найкращий метод. У більшості випадків простіше взяти файли пакунків, які вам необхідні, і скористатися із команди встановлення.

Команда **# apt-get install synaptic** встановить графічний інтерфейс для управління пакунками, разом із залежностями.

Деколи виникає потреба видалити непотрібний пакет з системи, щоб звільнити місце на жорсткому диску. Команда **#apt-get remove --purge назва\_пакету** видалить із системи відповідний пакет. Якщо цей пакет пов'язаний залежностями із іншими, а саме якщо його присутності потребує інша програма, то буде виведене повідомлення про порушення залежностей і запит на їхнє видалення.

У випадку встановлення (поновлення), чи видалення програмних пакунків, коли виникає конфлікт із залежностями, краще доставити всі потрібні пакети. В одній збірці **Linux** містяться всі необхідні файли для задоволення залежностей, і проблеми можуть виникнути лише при встановленні пакунків із зовнішніх джерел.

На перший погляд, система управління програмними пакунками **Apt** – це складна система, але завдяки своїй гнучкості вона дозволяє досконало управляти системою, автоматизувати процеси. Врешті, управління пакетами, очевидно, не є завданням щоденної потреби. Один раз ви вподобаєте собі набір пакетів, і досить довго не буде потреби будь-що міняти.

Загалом практичні рекомендації щодо роботи з пакунками для новачків такі:

під час встановлення використовуйте вже заготовлені набори, опираючись на які, в

режимі індивідуального вибору, сформуєте потрібний вам набір. Перегляньте перелік запронованих пакетів, ознайомтеся із вмістом збірки;

після встановлення випробуйте програми, визначтеся, що саме вам потрібне, доставте необхідне, заберіть зайве (якщо у вас бракує простору на ЖМД);

за переліком встановіть аналогічний набір на інші машини.

Після набуття певного практичного досвіду ви зможете створити систему автоматичного узгодження вмісту систем у мережі. На початковому етапі вам доведеться доставляти лише окремі пакети. У випадку класу, коли всі учнівські ПЕОМ однакові за обладнанням, можна також здійснити узгодження вмісту системи через мережу.

Зазвичай жодна із збірок **Linux** не може претендувати на те, що вона містить всі можливі програми, що розроблені для використання, навіть збірка **Debian**, що наразі охоплює понад 10000 програм. Проблема полягає у тому, що необхідно здійснити узгодження між пакетами, щоб задовільнити взаємозалежності між ними, і вирішувати її слід у рамках конкретної збірки **Linux**. Здійснюється це укладачами збірки, і спосіб, у який це зроблено, не обов'язково цілком тотожний у кожній збірці. Простіше кажучи, пакет у форматі **rpm** з однієї збірки не обов'язково встановиться коректним чином на іншій. Відкрите програмне забезпечення поширюється за приблизно такою схемою. Розробник оголошує про створення нової версії та розміщує архів із вихідним текстом програми у спеціальні репозиторії (архіви) в глобальній мережі Інтернет. Цей архів можна відвантажити з репозиторія, помістити у систему і скомпілювати його, потім встановити. Це доволі складний процес, до того ж компіляція достатньо великої програми може зайняти суттєвий проміжок часу. Зазвичай це виконують укладачі певної збірки **Linux**, які підтримують репозиторії з пакунками, що вже спаквані в певний формат, для конкретної системи управління програмними пакунками, для конкретної збірки.

Деколи розробник сам поширює вже скомпільовану програму у архіві, що супроводжується виконавчим файлом, який встановлює програму у систему. Таким чином переважно поширюються програми із закритими вихідними текстами, які походять від комерційних програм. Так, наприклад, поширюється офісний пакет **Open Office.org**, переглядач форматів **pdf Acrobat Reader** від фірми **Adobe** та інші. Встановлення таких пакунків відбувається таким чином:

розгортається архів;

запускається програма встановлення, яка міститься у ньому;

далі ця програма розгортає пакет у систему.

Для видалення постачають спеціальний файл-сценарій, якщо його немає, то видалення доведеться здійснювати вручну.

Інсталяційна збірка (дистрибутив) **Debian** є однією із найстаріших збірок **Linux**, разом із **Slackware**, і веде свій початок із 1994 року. Збірка **Debian** суттєво відрізняється від інших, переважно комерційних збірок, тим, що є громадською неприбутковою організацією, керується принципами Фонду Вільного Програмного забезпечення (**FSF**), ним же переважно і спонсорується. Саме тому вона має повну назву **Debian GNU/Linux**. Керівництво цим проектом здійснюється лідером, що обирається серед розробників. Завдяки такій схемі, яка надає змогу залучити до розробки і відлагодження дуже широке коло програмістів і адміністраторів, вдається узгоджено підтримувати в дистрибутиві понад 8000 пакунків. Жорсткі критерії оцінювання та перевірки програм забезпечують високу якість збірки, стабільність. Зворотним боком некомерційності **Debian** є часте відсування термінів остаточного випуску версій, порушення графіка, а також далеко не найсвіжіші версії ПЗ, яке потрапляє в остаточні випуски, а також гіршу підтримку драйверів комерційного походження.

Розробка **Debian** ведеться у трьох паралельних гілках:

стабільній (**stable**);

випробувальній (**testing**);  
нестабільній (**unstable**).

Стабільна гілка – це офіційно випущена версія разом із набором пакунків, надзвичайно докладно вивірена, до неї не вносяться нові версії програм, лише здійснюються виправлення помилок, які компрометують або можуть скомпрометувати систему з точки зору безпеки. До неї періодично додаються випуски (**releases**). Наприклад, стабільна версія 3.0 (кодова назва **Woody**), випуск 5.

Випробувальна гілка містить програмне забезпечення з версіями, що вийшли після випуску останньої стабільної версії і пройшли майже повне тестування. Склад випробувальної гілки змінюється доволі часто, але в принципі в якості некритичних щодо безпеки застосувань (наприклад, настільна система) випробувальна гілка пасує однозначно.

Нестабільна гілка містить пакунки, що вже пройшли узгодження із іншими пакунками та отримали початковий ценз. Склад нестабільної гілки змінюється дуже часто, її використання в повному обсязі може призвести до нестійкої системи.

Випуск нових версій здійснюється таким чином: стабільна гілка оголошується застарілою (хоча певний час ще здійснюється базова підтримка), випробувальна гілка стає стабільною і випускається. Нестабільна гілка переміщується на місце випробувальної розробки.

### 1.6.9. Клонування **Debian GNU/Linux**

Встановлення ОС **Debian GNU/Linux** на кілька однотипних ПЕОМ з оптичного диска є доволі тривалим процесом. Значно швидше можна виконати клонування **Debian GNU/Linux** (клонування ОС – це побайтне переписування інформації із одного ЖМД (або його окремих розділів) – джерела на інший диск-приймач. Для цього потрібно встановити у системний блок два ЖМД, один як **Master**, інший – як **Slave**. На рис.34 зображено роз'єднання ЖМД, а на рис. 35 – кабель для під'єднання двох ЖМД.

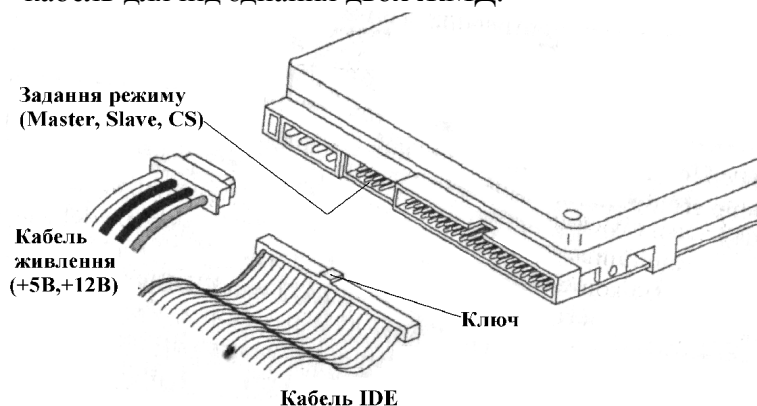


Рис. 34. Роз'єднання ЖМД

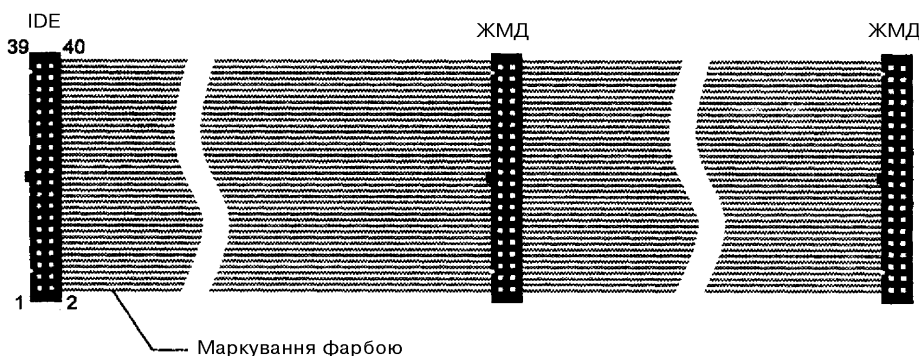


Рис. 35. Кабель IDE

Після встановлення двох ЖМД завантажте ОС **Linux** із диску, який ви хочете відклонувати (**Master**) і у терміналі від імені суперкористувача уведіть команду (приклад подано для випадку, коли диск із прототипом файлової системи розміщено головним (**master**) на **IDE1** першій шині, і вона відтворюється на головному диску (**master**) на **IDE2**):

```
# dd if=/dev/hda of=/dev/hdc
```

Після виконання цієї команди будемо мати два ЖМД з однаковим вмістом. Після перенесення відклоненого диска в ПЕОМ може виникнути проблема із завантажувачем ОС **GRUB**, який не перенісся коректним чином. Його слід відновити. Напростіший спосіб – скористатися можливостями самого **GRUB**. Для цього треба зробити два кроки:

1. створити дискету із завантажувальним записом **GRUB**;
2. завантажити за її допомогою ОС **Linux** на ПЕОМ із ушкодженим завантажувачем.

Створення дискети здійснюється від суперкористувача на справній системі, де встановлено будь-яку збірку **Linux** із завантажувачем **GRUB**. Слід увійти в командний рядок, вставити в пристрій чисту дискету, і виконати команду:

```
...#/sbin/grub-floppy /dev/fd0
```

Потім слід завантажитись із цієї дискети на тій ПЕОМ, яка потребує відновлення завантажувача. Якщо потрібно, змініть налаштування **BIOS**, щоб завантажитись з дискети. Після завантаження із дискети перед вами постане вбудований командний рядок **GRUB** із запрошенням **grub>**. У відповідь на це запрошення ви вводите команди, специфічні для **GRUB**, натисканням **Enter** запускаєте їх у дію.

Далі доведеться виконати таку послідовність дій:

1. Вказати, на якому із розділів, якого диска можна знайти ядро ОС **Linux**.

```
grub>root (hd0,0)
```

У цьому прикладі це головний (**master**) ЖМД на **IDE1** (hd0), перший розділ на ньому (0). Якщо ви ввели дані правильно, то отримаєте повідомлення, що виявлено розділ із файловою системою **ext3**, на розділі типу **Linux**;

2. Вказати на файл із ядром, і передати ядрові за допомогою опції, що початок дерева каталогів міститься на 00зділі **/dev/hda1** (ця інформація стосується Debian )

```
grub>kernel vmlinuz root=/dev/hda1 ro
```

а також вказати на файл із попередньо завантажуваними драйверами .

```
grub>initrd initrd.img
```

3. Завантажити систему, виходячи із даних, що були вказані вище.

```
grub>boot.
```

Після цієї команди завантажиться ядро ОС і ініціалізується повноцінна система. З наступного перевантаження ПЕОМ завантажувач сам відновиться. Якщо розбиття дисків на оригіналі і клоні сильно відрізняється, то слід виправити файл **/boot/grub/menu.lst**. Він містить послідовність команд, які виконує **GRUB**. Щоб кардинально перезаписати головний завантажувальний запис, слід виконати команду

```
...#/sbin/grub-install /dev/hda
```

Аналогічно відновлюється завантажувач **GRUB** і у випадку, коли під час встановлення якоїсь іншої ОС (наприклад, **MS Windows**) вона витерла завантажувач цілком. Під час операцій відновлення завантажувача слід бути дуже уважним і чітко розуміти структуру розділів і дисків вашої ПЕОМ.

Для реалізації цього методу потрібно встановити обидва ЖМД в один системний блок, що можливо лише тоді, коли системний блок не має гарантійних пломб. Якщо ж на системному блоці є гарантійні пломби, то клонування дисків можна здійснити по локальній мережі. Одним із методів клонування ОС по мережі є проект під назвою **SystemImager**.

Розглянемо встановлення і використання **SystemImager** під ОС **Debian GNU/Linux**. Для такого клонування потрібно встановити ОС **Debian GNU/Linux** із усім необхідним програмним забезпеченням на одну ПЕОМ (**golden client**). На іншу ПЕОМ також встановимо **Debian GNU/Linux**, можна з мінімальною конфігурацією програмного забезпечення (**image server**), Ця ПЕОМ буде зберігати і роздавати по мережі системний образ **golden client**'а іншим машинам. Після цього потрібно провести налаштування мережевих інтерфейсів ( опис у файлі `/etc/network/interfaces`) і встановити:

**Image Server** за допомогою команд

```
# apt-get update
```

```
apt-get install systemimager-server
```

**Golden Client** за допомогою команд

```
# apt-get update
```

```
# apt-get install systemimager-client
```

Після виконання цих дій переходимо до клонування:

1. На машині **golden client**'а із облікового запису **root** виконайте команду

```
# prepareclient --server image_server
```

де **image\_server** – IP-адреса або ім'я ПЕОМ **image server**'а.

Ця команда створить різноманітні файли в каталозі `/etc/systemimager` які будуть містити інформацію про розділи диска, типи файлових систем тощо. **prepareclient** також запустить демон **rsync**, який дозволить переправляти файли на ПЕОМ **image server**'а. Зараз **golden client** готовий передавати свій системний образ **image server**'у.

2. Тепер необхідно визначитися із методом призначення IP-адрес для ПЕОМ, на які ви будете копіювати образ **golden client**'а. Один із методів полягає в тому, що на машині **image server**'а ви запустите спеціальний сервіс **DHCP (Dynamic Host Configuration Protocol)**, який буде динамічно виділяти IP-адреси. Для налаштування цього сервісу під потреби **SystemImager** є спеціальна утиліта **mkdhcpserver**, яка після свого запуску буде вимагати у вас певну інформацію, необхідну для створення конфігураційного файлу **DHCP (/etc/dhcpd.conf)**. Інший метод (і значно легший) базується на тому, що уся необхідна мережева інформація буде поміщена в конфігураційний файл на дискеті. Ім'я цього файлу повинно бути **local.cfg** і він повинен знаходитися в кореневому каталозі дискети. Дискета повинна бути відформатована під файлову систему **ext2** або **fat**.

Розглянемо приклад файлу **local.cfg**.

```
HOSTNAME=host12
```

```
DEVICE=eth0
```

```
IPADDR=192.168.1.12
```

```
NETMASK=255.255.255.0
```

```
NETWORK=192.168.1.0
```

```
BROADCAST=162.168.1.255
```

```
IMAGESERVER=162.168.1.1
```

У полі **HOSTNAME** необхідно вказати ім'я ПЕОМ, на яку ви копіюєте системний образ; у полі **DEVICE** – активний мережевий контролер; у полях **IPADDR**, **NETMASK**, **NETWORK**, **BROADCAST** – IP-адресу активного мережевого інтерфейсу, маску мережі, адресу мережі і загальну адресу відповідно.

3. Виконайте команду **getimage**. Її базовий синтаксис такий:

```
# getimage -golden-client [client_hostname] -image [image_name]\
```

**getimage** зв'яжеться з **golden client**'ом і вимагатиме від нього файл



`/etc/systemimager/mounted_filesystems`, який містить список змонтованих файлових систем і їх точки монтування. **SystemImager** підтримує такі файлові системи, як **ext2**, **ext3** і **reiserfs**. Після того, як буде передано системний образ на **image server**, команда **getimage** автоматично згенерує сценарій клонування. Цей сценарій є **Bash**-сценарієм і його можна редагувати у будь-якому текстовому редакторі. Сценарій має розширення **master** і знаходиться у каталозі `/var/lib/systemimager/scripts`. містити інформацію про розділи диска, типи файлових систем тощо. Командою **addclients** встановить взаємозв'язок між іменами ПЕОМ і виконуваним інсталяційним **master**-сценарієм. Це інтерактивна програма, яка спочатку запитує діапазон імен ПЕОМ, на які буде встановлено заданий образ і відповідний їм діапазон **IP**-адрес.

4. Для старту клонування ви повинні завантажитися на відповідній машині з дискети, оптичного диска або мережі. Найпростіше буде завантажити машину із спеціальної дискети. Для створення такої дискети цього вставте у ГМД (на **image server**'і) пусту дискету і запустіть команду

```
# mkautoinstalldiskette /dev/fd0
```

Ця команда створить дискету з ядром **Linux** і всіма необхідними для завантаження файлами. Створіть файл **local.cfg** у кореневому каталозі цієї дискети і внесіть в нього всю необхідну мережеву інформацію. Після цього вставте виготовлену дискету у ГМД відповідної ПЕОМ і ввімкніть її. Внаслідок завантаження з дискети буде налаштований відповідний мережевий інтерфейс згідно файлу **local.cfg** і ПЕОМ буде готова прийняти і виконати **master**-сценарій із ПЕОМ **image server**'а. Цей сценарій виконає копіювання образу системи на ЖМД цієї ПЕОМ.

Зауваження: Магнітний диск **image server**'а повинен мати вільне місце для збереження системного образу **golden client**'а. Усі ПЕОМ, що будуть використовувати цей образ, повинні бути максимально подібними. Це означає, що вони повинні мати однакові набори системної логіки (**chip set**) і однаковий тип (наприклад, **IDE**, **SCSI**, **Mylex Hardware RAID** тощо).

## 2. Прикладне програмне забезпечення для **IBM**-подібних ПЕОМ

Саме наявність текстового редактора “**Електричний олівець**” і табличного процесора “**VisiCalc**” у програмному забезпеченні персонального комп'ютера **Apple II** були причиною широкого поширення **Apple II** в офісах американських фірм. Після появи **IBM**-подібних ПЕОМ багато програмістів і фірм долучилося до створення програм, які в основному орієнтувались на офісні операції:

- уведення тексту, зберігання уведеного тексту на магнітних дисках і роздрук його на аркушах паперу;

- опрацювання числової інформації в електронних таблицях;

- уведення, зберігання і пошук числової і текстової інформації у базах даних.

Такі комплекси програм стали називати офісними пакетами. Доволі швидко лідером у цій області стала фірма **Microsoft** з своїм офісним пакетом **Microsoft Office**. Слід наголосити на існуванні і інших офісних пакетів – **Lotus Office**, **Star Office**, **Open Office**, **АБ-офіс** (перелік неповний). На думку авторів, причиною широкого поширення в Україні **Microsoft Office xx** є не його переваги над іншими офісними пакетами, а задовнена практика користування піратськими копіями **Microsoft Office** (довгий час фірма **Microsoft** “*не помічала*” комп'ютерного піратства в Україні) і інерція мислення користувачів ПЕОМ (навіщо освоювати безоплатний і маловідомий **OpenOffice.org**, коли можна встановити крадену версію широко відомого **Microsoft Office xx**). Після появи у Кримінальному Кодексі України статті 176 “Про охорону авторських і суміжних прав” українські користувачі ПЕОМ

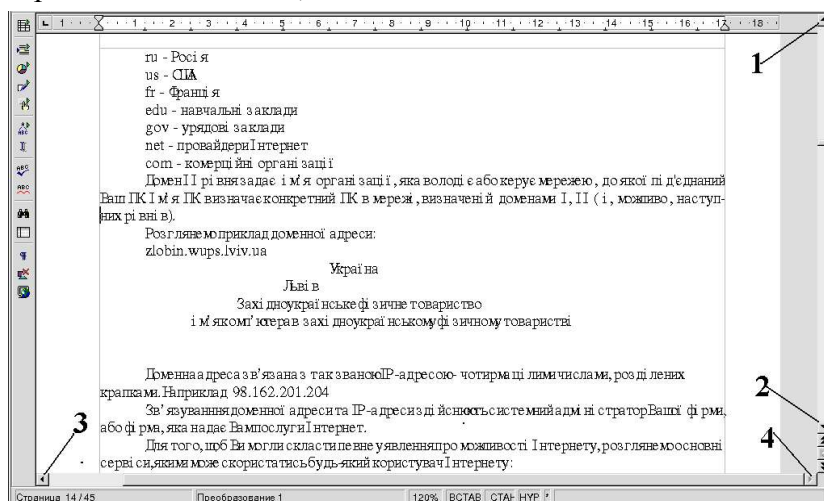
постали перед вибором – купувати широко відомий і доволі дорогий **Microsoft Office xx**, користуватись краденою версією **Microsoft Office xx** чи встановити вільно-поширюваний **OpenOffice.org**.

## 2.1 Офісний пакет **OpenOffice.org**

Офісний пакет **OpenOffice.org** є відкритою версією офісного пакету **Star Office**, який належить фірмі **Sun**. Створено версії **OpenOffice.org** як для ОС **Linux**, так і для ОС **Microsoft Windows xx**. Розглянемо складові цього пакету.

### 2.1.1 Текстовий процесор **OpenOffice.orgWriter**

Після появи персональних комп'ютерів опрацювання текстової інформації на ПЕОМ дуже швидко стало найбільш масовим завданням. Кількість програм, які виконують опрацювання текстів, надзвичайно велика. Найбільш поширеними серед них є текстові



редактори, текстові процесори та видавничі системи. У роботі над текстовим документом текстовий редактор схожий на вікно, через яке людина проглядає текст.

Рис. 36. Вікно текстового редактора з органами керування

1,2 - піктограми переміщення вікна по тексту у вертикальному напрямку (вертикальне прокручування), 3,4 - піктограми переміщення вікна по тексту у горизонтальному напрямку (горизонтальна прокручування)

У вікні можна вводити новий текст, змінювати (редагувати) вже створений текст. Спеціальний вказівник (курсор) - вертикальна риска, яка періодично з'являється на екрані, показує місце у тексті, в якому можна вводити, видаляти або вставляти текст. При цьому текстовий редактор може працювати у режимі вставлення (**Insert - ВСТАВ**) тексту, або у режимі "писання поверх тексту" (**Overwrite - ВИЩЕ**). Перемикання режимів здійснюється клавішею **Insert**. Для переміщення у вікні використовують клавіші керування курсором

Рис. 37. Клавіші керування курсором

1,2 - переміщення курсора по тексту у горизонтальному напрямку, 3,4 - переміщення курсора по тексту у вертикальному напрямку, 5 - переміщення по тексту на сторінку вгору (**Page Up**), 6 - переміщення по тексту на сторінку вниз (**Page Down**), 7 - "перескок" курсора на початок рядка (**Home**), 8 - "перескок" курсора на кінець рядка (**End**), 9 - перемикання режиму написання тексту (**Insert/Overwrite**), 10 - клавіша витирання (видалення) символу справа від курсора

Якщо курсор розташований у крайніх позиціях вікна, то натискання клавіш керування курсором приводить до переміщення вікна у напрямку, що визначається натиснутою клавішею. Для швидкого переміщення вікна по тексту можна використати клавіші **Page Up**, **Page Down**, **Home**, **End**. Набір тексту здійснюють за допомогою основної клавіатури

### Рис. 38. Афавітно-цифрова частина клавіатури

1- витирання символів зліва від курсора, 2 – завершення вводу рядка тексту, 3 - пропуск однієї позиції у тексті (прогалина), 4 – перемикання регістра вводу (верхній/нижній), 5 – фіксація регістра, 6 – табуляція

Перемикання алфавіту (латинський, український, російський і т.д) виконують за допомогою комбінацій клавіш **Ctrl Shift**, або **Alt Shift**, або **Shift Shift** (залежно від налаштування операційної системи) або лівою клавішею „мишки”. Знаки кирилиці на клавіатурі позначаються, як правило, червоним кольором, латинки - чорним. У верхньому ряду основної клавіатури розміщені клавіші з позначеннями цифр (нижній регістр) та допоміжних знаків (верхній регістр). При наборі великої кількості цифрової інформації доцільно користуватись цифровою клавіатурою

### Рис. 39. Цифрова клавіатура

1 - перемикання режимів цифрової клавіатури : набір цифр - світлодіод **Num Lock** світиться, керування курсором - світлодіод **Num Lock** не світиться, 2 - світлодіоди

Існують два види програм для роботи з текстом. Перші – це текстові редактори, які працюють з текстом як набором кодів знаків. Вони не змінюють вигляду тексту (вигляду знаків, форматування документа) і використовуються для редагування текстів програм і сценаріїв роботи програм. Другі – це текстові процесори, які змінюють розмір і стиль написання знаків, форматування документа, дозволяють вставляти у текстовий документ таблиці і графічні зображення. У текстових процесорах реалізовано принцип „що бачиш, те й отримаєш”.

#### 2.1.1.1 Уведення тексту

Запустити текстовий процесор **OpenOffice.orgWriter** можна через стартове меню графічної оболонки або запускателем із панелі інструментів стільниці. Після запуску **OpenOffice.orgWriter** на екрані дисплею розкриється таке вікно

### Рис. 40. Вікно текстового процесора **OpenOffice Writer**

1 - заголовок вікна, 2 - рядок меню програми, 3 – лінійка інструментів, 4 - вікно тексту, 5,6 - лінійки зміщення вікна тексту у вертикальному та горизонтальному напрямках, 7 - піктограма закінчення роботи програми, 8 - піктограма зміни розмірів вікна програми, 9 - піктограма згортання вікна програми у значок на лінійці стану робочого стола, 10 - рядок стану вікна програми, 11 - піктограма перемикання режимів **Insert** (додавання нових символів у текст)/**Overwrite** (написання нових символів поверх існуючого тексту), для встановлення потрібного режиму використовують клавішу **Insert** на клавіатурі.

Для створення або відкриття документа **OpenOffice.org** виберіть пункти меню:

“**Файл – Новий – Текстовий документ**” (“**File - New - Text Document**”) - створення нового документа;

“**Файл – Відкрити**” (“**File - Open**”) - відкриття існуючого документа.

На рис. 41 подана копія екрану пакету **OpenOffice.org** при виборі пункту створення документа.

### Рис. 41. Вікно пакету **OpenOffice.org** при створенні нового документа

Після цього ви можете вводити новий текст з клавіатури (створення нового текстового документа) або вносити зміни (редагувати) у вже існуючий документ. Для оформлення тексту (вибір шрифту, його написання, розміру, розташування на сторінці) можна скористатись з лінійки інструментів текстового процесора **OpenOffice.orgWriter**

### Рис. 42. Лінійка інструментів текстового процесора **OpenOffice.orgWriter**

1 - вибір шрифту, 2 - вибір розміру шрифту, 3 - задання жирності символу (натиснуто - жирний шрифт, відтиснуто - звичайне написання), 4 - написання символів курсивом, 5 - вибір шрифту з підкресленням, 6 - задання розміщення тексту на сторінці

(вирівнювання зліва, центрування, вирівнювання справа, вирівнювання з обох сторін), 7 - задання методу виділення елементів списку

Якщо вам потрібно змінити написання фрагмента тексту (одного або кількох слів, кількох речень, кількох абзаців), виділіть цей фрагмент (встановіть курсор на початок фрагменту і, затиснувши ліву клавішу "мишки", "перетягніть" курсор до кінця фрагменту - фрагмент перейде в інверсне зображення [білі букви, чорне тло]). Після цього „виберіть мишкою” піктограми панелі інструментів, які задають потрібне вам оформлення тексту - оформлення виділеного фрагменту тексту змінюватиметься відповідно до вказаних вами кнопок панелі інструментів процесора **OpenOffice.orgWriter**. Наберіть який-небудь текст (хоча б цей абзац) і поекспериментуйте з панеллю інструментів текстового процесора. Якщо в процесі набору тексту ви пропустили символ або слово, перемістіть курсор у потрібне місце тексту і у режимі **Insert** введіть потрібний символ або слово. виправлення неправильно набраних символів доцільно здійснювати у режимі **Overwrite** - встановіть потрібний режим, перемістіть курсор у потрібне місце тексту і введіть потрібний символ - він буде написаний (писання поперх існуючого тексту) замість неправильно набраного символу.

Після завершення набору нового тексту його потрібно зберегти на магнітному диску.

### 2.1.1.2 Збереження тексту на магнітних дисках та відкриття існуючого текстового файлу.

Для збереження набраного тексту на магнітному диску виберіть пункти меню:

“**Файл-Зберегти**” (“**File – Save**”) - для збереження уже існуючого текстового файлу;

“**Файл-Зберегти як**” (“**File – Save As**”) - для збереження новоствореного тексту або уже існуючого текстового файлу з іншим іменем.

Рис. 43. Вікно збереження тексту на магнітному диску

1 - відображення вмісту вибраного каталога, 2 - створення нового каталога, 3 - швидкий перехід у домашній каталог, 4 - перехід у каталог вищого рівня, 5 -рядок задання імені файлу, 6- рядок (випадаючий список) вибору розширення текстового файлу (**sxw** - власний формат **OpenOffice Writer**, **rtf** - формат, який використовується багатьма текстовими процесорами [ у т.ч. **MS Word**], **html**, **text Unix**, **text Mac**, **text Dos**, формат **MS Word**)

Відкриття існуючого текстового файлу є протилежною операцією по відношенню до операції збереження файлу. На рис. 44 подано вікно відкриття існуючого файлу (це вікно має один і той же вигляд у всіх компонентах пакету).

Рис. 44. Вікно відкриття існуючого файлу

1 - відображення вмісту вибраного каталога, 2 - створення нового каталога, 3 - швидкий перехід у домашній каталог, 4 - перехід у каталог вищого рівня, 5 - рядок задання імені файлу, 6- рядок (випадаючий список) вибору розширення текстового файлу (**sxw** - власний формат **OpenOffice Writer**, **rtf** - формат, який використовується багатьма текстовими процесорами [ у т.ч. **MS Word**], **html**, **text Unix**, **text Mac**, **text Dos**, формат **MS Word**)

### 2.1.1.3 Друк текстового документа.

Якщо ви закінчили оформлення текстового документа, то, можливо, вам потрібно його роздрукувати. Для цього увійдіть у пункт меню “**Файл**” (“**File**”)

Рис. 45. Група підпунктів меню “**File**” (“**Файл**”), які призначені для друку документа

1 - попередній перегляд сторінки, 2 - відкриття діалогового вікна друку, 3 - налаштування пристрою друку

Перш ніж роздрукувати документ, доцільно переглянути його розміщення на

аркушах паперу. Якщо розміщення тексту на листках паперу вас влаштовує, переходьте до друкування документа. На рис. 46 подано діалогове вікно “Друк”(“Print”) пакету **Open Office.org**.

Рис. 46. Діалогове вікно “Друк” пакету **Open Office.org**.

1 - рядок вибору пристрою друку, 2 - рядок задання діапазону друку, 3 - рядок задання кількості копій, 4 - відкриття діалогового вікна “Властивості” (“Properties”) - у цьому вікні задаються формат паперу та орієнтація тексту на листку паперу, властивості друку графічних зображень, чорновий чи висодоякісний друк

Після задання необхідних параметрів (або прийняття параметрів “за замовчуванням”) вам залишається „вказати мишкою” на “ОК” і дочекатися завершення друку. Дуже корисною функцією текстового процесора **OpenOffice.orgWriter** є друк брошур - кожні дві сторінки зменшуються і роздруковуються на одному листку паперу в альбомному форматі, наступні дві сторінки роздруковуються на звороті листка паперу. Роздрукований документ потрібно скласти, прошити по лінії розділу, перегнути - і ви отримали готову книжечку. На рис. 47 подано діалогове вікно “Параметри” (“Printer Options”), за допомогою якого можна задати друк брошури.

Рис. 47. Діалогове вікно “Параметри”

#### 2.1.1.4 Пошук заданої інформації

Завдяки пошуковій функції текстового процесора **OpenOffice.orgWriter** ви можете швидко знайти певний текст у текстовому документі. Для запуску пошуку вам потрібно у пункті меню **OpenOffice.orgWriter** “Правка” (“**Edit**”) вибрати пункт “Знайти і замінити” (“**Find&Replace**”).

Рис. 48. Зміст пункта меню “Правка”

У діалоговому вікні “Знайти і замінити” (“**Find&Replace**”) потрібно задати зразок та опції пошуку і натиснути потрібну кнопку “Знайти” (“**Find**”) - до першого зразка у тексті або “Знайти все” (“**Find all**”) - послідовний пошук заданого зразка у всьому тексті.

Рис. 49. Діалогове вікно “Знайти і замінити” у режимі пошуку

Для автоматичної заміни заданого тексту потрібно задати зразки пошуку та заміни.

### 2.1.1.5 Робота з таблицями

Табличну форму подання інформації широко використовують у ділових документах, для поєднання в одному документі тексту і таблиць можна скористатись з пункту меню текстового процесора **OpenOffice.orgWriter** “Вставка” (“Insert”)

Рис. 50. Пункт “Вставка” меню текстового процесора **OpenOffice.orgWriter**

Після того, як ви “вказете” мишкою на пункт “Таблиця” (“Table”) (або одночасно натиснете клавіші **Ctrl F12**) на екрані дисплея з’явиться діалогове вікно “Вставити таблицю” (“Insert Table”).

Рис. 51. Вікно задання параметрів таблиці

1 - ім'я таблиці, 2 - кількість стовпців, 3 - кількість рядків

Після заповнення полів цього вікна у документі з’явиться заготовка таблиці з заданою кількістю рядків і стовпців, а до панелі інструментів **OpenOffice.orgWriter** додається панель інструментів для роботи з таблицями

Рис. 52. Панель інструментів для роботи з таблицями процесора **OpenOffice.orgWriter**

1 - сумування виділеної області таблиці, 2 - задання розмірів полів таблиці, 3 - об’єднання виділених комірок, 4 - розбиття виділеної комірки, 5 - вставка рядка, 6 - вставка стовбця, 7 - видалення рядка, 8 - видалення стовбця, 9 - обрамлення таблиці, 10 - задання типу ліній, 11 - задання кольору фону комірок таблиці.

### 2.1.1.6. Робота з графікою

Текстовий процесор **OpenOffice.orgWriter** дозволяє вставляти у текстовий документ графічні зображення, для цього потрібно вибрати відповідний пункт у меню “Вставка” (“Insert”)

Рис. 53. Пункти меню “Вставка - Графіка”

Графічне зображення можна отримати з файлу або пристрою сканування зображень (“Пошук”). Разом з пакетом **Open Office.org** ви отримуєте і велику кількість графічних файлів, які розміщені у каталозі `/opt/OpenOffice.org1.0/share/gallery/`. Їх можна використати для оформлення своїх текстових документів. На рис. 54 подано діалогове вікно вибору графічного файлу для вставлення його у текстовий документ

Рис. 54. Діалогове вікно вибору графічного файлу

1- список файлів, 2- вікно перегляду вибраного файла, 3 - швидкий перехід у домашній каталог, 4 – створення нового каталога, 5 – перехід у каталог (папку) вищого рівня, 6 -рядок задання імені файлу, 7-рядок (випадаючий список) вибору розширення графічного файлу

Взаємне розташування тексту і графічних образів задається за допомогою контекстного меню - “вказіть” на малюнок лівою клавішею "мишки" (довкола малюнка з’явиться прямокутна рамка з точками деформації) і натисніть праву клавішу "мишки". На екрані відкриється контекстне меню задання взаємного розміщення тексту і графічних образів

Рис. 55. Контекстне меню задання взаємного розміщення тексту і графічних образів

Після вибору пункту меню “Обгортка” (“Wrap”) ви можете задати потрібне для вам розташування тексту і графіки у текстовому документі. Окрім вбудовування в текстовий документ графічних зображень, які отримані за допомогою інших програм (див. Опрацювання графічної інформації), текстовий процесор **OpenOffice.orgWriter** має власні засоби малювання, які можна викликати „вказуванням” на піктограму Після цього на екрані дисплея з’явиться панель функцій малювання



**Рис. 56 Панель функцій малювання процесора OpenOffice.orgWriter**

1 - вибір об'єкта, 2- лінія, 3 - прямокутник, 4 - еліпс, 5 - багатокутник, 6 - крива Без'є, 7 - мальована форма, 8 - дуга, 9 - сегмент еліпса, 10 - сектор дола, 11 - текст, 12 -біжучий рядок, 13- легенда

Після вибору потрібного об'єкта панель інструментів процесора **OpenOffice.orgWriter** доповниться панеллю графічних побудов

**Рис.57. Панель інструментів графічних побудов процесора OpenOffice.orgWriter**

1 - редагування точок, 2- лінія, 3 - стиль закінчення ліній, 4 - випадаючий список стилів ліній, 5 - задання товщини лінії, 6 - випадаючий список кольорів ліній, 7 – заливка замкнутої області, 8 - випадаючий список кольорів областей, 9 - випадаючий список стилю/кольору заповнення фігур (прямокутник, еліпс і т.ін.), 10 - обертання об'єкта, 11 - зміна точки прикріплення об'єкта до текстового документа, 12 - розташування об'єкта на передньому плані, 13 - розташування об'єкта на задньому плані, 14 – пересунути на передній план, 15 – послати назад, 16 – упорядкувати об'єкт

Малюнки складаються з окремих елементів, які можна переміщувати один відносно одного, для цього достатньо натиснути кнопку вибору об'єкта і, після цього, вказати мишкою на об'єкт. На об'єкті з'являться граничні відмітки у вигляді квадратиків. Після цього виділений об'єкт можна переміщувати, розтягувати, стискати, видаляти. На рис. 58 подано малюнок з виділеним елементом

**Рис. 58. Малюнок з виділеними елементами**

Для того, щоб малюнок, як множина з окремих елементів, не розпадався на частини при редагуванні документа, його потрібно об'єднати в одне ціле. Це можна здійснити таким чином:

виберіть інструмент відмітки об'єкта;  
виділіть малюнок лівою клавішею "мишки";  
натисніть праву клавішу "мишки" і у контекстному меню виберіть пункт  
**"Групувати" ("Group")**.

**Рис. 59. Згрупований малюнок**

До коротких текстів (заголовки, девізи, емблеми) можна застосовувати графічні ефекти. Для цього потрібно:

відкрити функції малювання;  
вибрати функцію вводу тексту;  
намалювати текстову рамку;  
ввести текст;  
у пункті меню **"Формат" ("Format")** вибрати **"Шрифтові ефекти" ("FontWork")**;  
вибрати один із пропонованих графічних ефектів.

На рис. 60 подано вікно шрифтових ефектів, а на рис. 61 приклад написання тексту півколом

Рис. 60. Вікно графічних ефектів “Шрифтові ефекти”

Рис. 61. Написання тексту півколом

Створений об’єкт можна трансформувати (деформувати) шляхом перетягування контрольних точок об’єкта.

Текст можна розміщувати по мальованій лінії, для цього потрібно:

намалювати лінію;

двічі “клацнути” по ній лівою клавішею "мишки" і ввести текст;

у пункті меню “Формат” (“Format”) вибрати “Шрифтові ефекти” (“FontWork”).

На рис. 62 подано приклад написання тексту з нахилом

Рис. 62. Написання тексту з нахилом

### 2.1.2 Табличний процесор **OpenOffice.org.Calc**

Електронні обчислювальні машини (ЕОМ) створювались у першу чергу для виконання обчислень (опрацювання числової інформації). Сучасне програмне забезпечення надає користувачу ПЕОМ широкі можливості в галузі числової обробки інформації. Для опрацювання числової інформації у прямокутних таблицях використовують табличні процесори. Електронна таблиця складається з окремих комірок (чарунк), які утворені перетином стовбців (A,B,C,D,E,F,...Z,AA,...AZ,BB...BZ,CC...CZ,DD...DZ,...ZZ ) та рядків (1,2,3,4,5,... 32000 рядків). Таким чином конкретна комірка позначається A1, D7, AA17. У комірку можна ввести:

текст - використовують для створення пояснювальних написів;

числове значення - основний вид інформації в електронних таблицях;

формулу - використовують для перетворення інформації.

Числова інформація може подаватись у кількох форматах - звичайному (ціла частина, дробова частина), з “плаваючою комою”, грошовому, відсотків, дати і часу.

Над рядками або стовбцями електронної таблиці можна проводити обчислення, які визначаються можливостями конкретної програми опрацювання електронних таблиць. На рис. 63 подано вікно програми опрацювання електронних таблиць пакету **OpenOffice.org** з заповненими стовпцями A-Y та рядками 1-6.

Рис. 63. Вікно табличного процесора із заповненими комірками (чарунками)

1 - меню табличного процесора, 2 - панель інструментів табличного процесора, 3 - лінійка вертикального прокручування, 4 - лінійка горизонтального прокручування, 5 - вікно відображення імені активної комірки, 6 - вікно відображення формули активної комірки, 7 - текстова інформація в комірці, 8 - числова інформація в комірці, 9 - активна комірка, 10 - ім'я листа, 11 - автопілот функцій, 12 - службовий рядок табличного процесора

Інформацію, яка введена у таблицю у цьому прикладі, можна розділити на текстову і числову (цілу частину відділяють від дробової десятковою комою!). Окрім текстової і числової інформації у комірки електронної таблиці можна увести формульну інформацію. Розберемо правила вводу інформації у комірки таблиці. Насамперед для вводу інформації у потрібну комірку потрібно “вказати” на неї “мишкою”. Після цього комірка буде виділена чорною рамкою (комірка G2 на рис. 63). Текстова інформація вводиться з використанням основної групи клавіш клавіатури, числова - з використанням цифрових клавіш, ціла частина відділяється від дробової комою. Якщо ви ввели текст довший, ніж ширина комірки, частина тексту для вас буде невидимою. Подвійне “клацання” лівою клавішею "мишки" на розділовій вертикальній лінії встановить “оптимальну” ширину стовбця. Якщо ширина введеного або обчисленого числового значення більша ширини комірки, то замість числа ви

побачите символи ### - встановіть “оптимальну” ширину стовбця і число буде повністю відображено у домірці. Ввід формули починається знаком =. У формулах можна використовувати стандартні функції табличного процесора, виклик стандартних функцій доцільно здійснювати за допомогою асистента функцій (11 на рис.63). На рис. 64 подано вікно асистента функцій з логічною функцією обчислення внеску у пенсійний фонд (якщо заробітна плата складає не більше 150 грн./місяць, то внесок у пенсійний фонд складає 1%, більше 150 грн./місяць – 2%.

Рис. 64. Вікно асистента функцій табличного процесора

1 - випадаючий список груп функцій, 2 - вікно вибору функції, 3 - вибрана функція, 4 - вікно відображення введеної формули, 5 – логічний вираз, 6 – формула для обчислення значення за умови істинності логічного виразу, 7 - формула для обчислення значення за умови хибності логічного виразу

Стандартні функції табличного процесора об’єднані в групи (назви груп подаються так, як вони відображаються асистентом функцій у **OpenOffice.org 1.1.0**):

- усі (повний алфавітний список);
- база даних;
- дата і час;
- фінансовий;
- інформація;
- логічний;
- математичний;
- масив;
- статистичний;
- таблиця;
- текст;
- додаток.

Отримати опис стандартної функції можна за допомогою довідки табличного процесора.

Процесор **OpenOffice.orgCalc** дозволяє сортувати інформацію у таблицях, для цього потрібно виділити прямокутну область у таблиці, в якій ви хочете здійснити сортування інформації та у пункті меню “Дані” (“**Data**”) вибрати підпункт “Сортування” (“**Sort**”). У вікні “Сортування” (“**Sort**”) потрібно задати стовбець таблиці і напрямок сортування (по зростанню чи по спаданню). Сортування проводиться за алфавітом, якщо заданий стовбець є текстовим, або за числовим значенням, якщо заданий стовбець є числовим. На рис. 65 подано пункт меню “Дані”, а на рис. 64 - вікно задання параметрів сортування

Рис. 65. Пункт меню “Дані”

Рис. 66. Вікно задання параметрів сортування

Інформацію в електронній таблиці можна розміщувати на кількох аркушах, “зшитки” з кількох аркушів називаються книгами. Для адресації комірок по всій книзі до імені комірки додається ім’я аркуша, повне ім’я комірки виглядає так:

**Ім’я аркуша. ім’я комірки.**

Ім’я аркуша можна змінити (стандартно Аркуш1, Аркуш2 і т.д.). Для цього потрібно “вказати” мишкою на ім’я аркуша і викликати контекстне меню натисканням правої клавіші “мишки”. На рис. 67 подано контекстне меню операцій з аркушами електронної книги

Рис. 67. Контекстне меню операцій з аркушами

Досить часто для якісної оцінки ряду чисел використовують графіки та діаграми. Табличний процесор має потужні засоби побудови графіків та діаграм. На рис. 68 поданий приклад побудови колової діаграми на основі ряду чисел листка “податки”, які дозволяють оцінити частку дожного показника у загальній сумі коштів на оплату праці працівників підприємства. Для побудови графіків і діаграм потрібно скористатись автопілотом побудови діаграм “Вставка -Діаграма” (“Insert - Chart”). На рис. 69 подані вікна автопілота побудови діаграм табличного процесора

Рис. 68. Аркуш електронної таблиці з коловою діаграмою

Рис. 69. Вікна автопілота побудови діаграм (кроки 1 та 2)

Інформацію, внесену у електронні таблиці, ви можете зберігати, роздруковувати на папері, редагувати і т.ін. Ці дії виконуються так само, як і дії по збереженню, друку, редагуванню текстових документів.

### 2.1.3 Редактор векторної графіки **OpenOffice.org.Draw**

Після запуску **OpenOffice.org.Draw** на екрані дисплея відкриється вікно

Рис. 70. Вікно редактора векторної графіки **OpenOffice.org.Draw**

Розглянемо призначення окремих піктограм панелі інструментів редактора **OpenOffice.org.Draw**

- інструмент виділення прямокутної області зображення (виділення штриховою лінією). Над виділеним фрагментом можна проводити такі дії - вирізання, копіювання (для подальшого вставлення у зображення - пункт меню "Правка"), видалення частини зображення (клавішею **Delete**), групування зображення.

- інструмент збільшення зображення. Для збільшення зображення потрібно "навести" вказівник "мишки" на зображення і натиснути ліву клавішу "мишки". Для зменшення зображення потрібно натиснути клавіші **Ctrl -**.

- інструмент вводу надписів на об'єктах. Після вибору цього інструмента потрібно "відмітити мишкою" прямокутну область надпису, вибрати шрифт та колір символів і ввести текст надпису.

- інструмент побудови прямокутних областей.

- інструмент побудови кіл та еліпсів.

- інструмент побудови об'ємних фігур.

- інструмент побудови кривих.

- інструмент побудови прямих ліній.

- інструмент побудови ламаних ліній.

- інструмент обертання об'єкта довкола заданої осі обертання. Положення осі обертання об'єкта задається натисканням лівої клавіші "мишки", після чого ви можете обертати об'єкт, "захопивши" одну з його точок лівою клавішею "мишки".

При формуванні зображення у редакторі **Open Draw** ви можете змінювати (збільшувати/зменшувати) розміри об'єктів, переміщувати їх у межах зображення, для цього потрібно навести вказівник "мишки" на об'єкт і натиснути ліву клавішу "мишки", довкола об'єкта з'явиться рамка з маленькими квадратами у точках зміни розмірів об'єкта

Рис. 71. Приклади графічних об'єктів з виділенням

Після завершення роботи над зображенням його потрібно зв'язати в одне ціле - згрупувати об'єкти. Для цього потрібно виділити зображення, далі вибрати в меню редактора пункти "**Дії - Групувати**". Після цього всі елементи, які входять у виділену область, будуть об'єднані в одне ціле. Над групованим зображення можна виконувати операції переміщення, копіювання, видалення і масштабування.

Створене зображення можна зберегти на магнітному диску, роздрукувати на пристрої друку. Дії по збереженню або друку зображення дуже подібні до таких же дій з текстовими

документами.

#### 2.1.4. Система побудови презентацій **OpenOffice.org.Impress**

Системи побудови презентацій дозволяють поєднати в одному документі текст, графічні зображення і звукову інформацію. Показ презентації на великому екрані за допомогою відеопроектора дозволяє подати інформацію, внесenu у презентацію, великій аудиторії. Розглянемо методи побудови презентацій.

Для створення нової презентації виберіть **“Файл – Новий - Презентація”** (**“File - New - Presentation”**) - **OpenOffice.org** запустить автопілота побудови презентацій. На рис. 72 подано вікно перше вікно асистента побудови презентацій

Рис. 72. Перше вікно асистента побудови презентацій

Виберіть **“Порожня презентація”** (**“Empty presentation”**) і **“натисніть”** мишкою **“Далі”** (**“Next”**). У другому вікні автопілота (рис. 73) задайте носій на якому буде відображатись презентація (екран дисплея, папір, слайд, ускладнений аркуш [ прозїрка]) і тло слайда та **“натисніть”** мишкою **“Далі”** (**“Next”**).

Рис. 73. Друге вікно асистента побудови презентацій

У третьому вікні асистента (рис. 74) задайте метод показу слайдів і **“натисніть мишкою”** кнопку **“Створити”** (**“Create”**) - відкриється вікно задання розмітки слайда (рис. 75).

Рис. 74. Третє вікно асистента побудови презентацій

Рис. 75. Вікно задання розмітки слайда

Виберіть бажану для вас структуру слайду і заповніть його поля текстом і графічними зображеннями. На рис. 76 подано вікно системи побудови презентації із слайдом презентації, а на рис. 77 – це й же слайд у режимі демонстрації

Рис. 76. Вікно системи побудови презентації із створеним слайдом презентації

Рис. 77. Створений слайд у режимі показу

Після заповнення першого слайду додайте по черзі наступні слайди. По завершенню роботи над презентацією або її частиною збережіть побудовану презентацію у файлі - **“Файл-Зберегти”** (**“File Save”**). Якщо ви плануєте показувати презентацію з прозїрок або з аркушів паперу – роздрукуйте її на пристрої друку.

#### 2.2. Офісний пакет **GNOME Office**

До складу **GNOME Office** наразі входять лише дві програми - текстовий процесор **Abiword**, табличний процесор **Gnumeric**. Розглянемо їх

## 2.2.1. Текстовий процесор **Abiword** 2.2.1.1. Уведення тексту в ПЕОМ

Після запуску текстового процесора **Abiword** відкриється вікно

Рис. 78. Текстовий процесор **Abiword**

Для створення або відкриття документа виберіть пункти меню:

“**Файл – Новий**” - створення нового документа;

“**Файл – Відкрити**” - відкриття існуючого документа.

На рис. 79 подані копії екранів процесора **Abiword** при виборі пункту створення і відкриття документа

а)

б)

Рис. 79. Вікна **Abiword** при створенні нового документа (а) і відкритті існуючого документа (б).

Після цього ви можете вводити новий текст з клавіатури (створення нового текстового документа) або вносити зміни (редагувати) у вже існуючий документ. Для оформлення тексту (вибір шрифту, його написання, розміру, розташування на сторінці) можна скористатись з лінійки інструментів процесора **Abiword**

Рис. 80. Лінійка інструментів процесора **Abiword**

Якщо ви потребуєте змінити написання фрагмента тексту (одного або кількох слів, кількох речень, кількох абзаців), виділіть цей фрагмент (встановіть курсор на початок фрагменту і, затиснувши ліву клавішу "мишки", “перетягніть” курсор до кінця фрагменту – фрагмент перейде в інверсне зображення [білі букви, синє тло]). Після цього „виберіть мишкою” піктограми панелі інструментів, які задають потрібне вам оформлення тексту – оформлення виділеного фрагменту тексту змінюватиметься відповідно до вказаних вами кнопок панелі інструментів процесора. Наберіть який-небудь текст (хоча б цей абзац) і поекспериментуйте з панеллю інструментів текстового процесора. Якщо в процесі набору тексту ви пропустили символ або слово, перемістіть курсор у потрібне місце тексту і у режимі **Вставка** введіть потрібний символ або слово. виправлення неправильно набраних символів доцільно здійснювати у режимі **Заміщення** – встановіть потрібний режим, перемістіть курсор у потрібне місце тексту і введіть потрібний символ – він буде написаний (писання поверх існуючого тексту) замість неправильно набраного символу. Після завершення набору нового тексту його потрібно зберегти на магнітному диску.

### 2.2.1.2. Збереження тексту на магнітних дисках

Для збереження набраного тексту на магнітному диску виберіть пункти меню:

“**Файл - Зберегти**” – для збереження уже існуючого тексту;

“**Файл - Зберегти як**” – для новоствореного тексту (або для збереження уже існуючого текстового файлу з іншим іменем).

Рис. 81. Вікно збереження тексту на магнітному диску

1 - навігація по файлової системі, 2 - рядок задання імені файлу, 3 - випадним списком детермінованих каталогів, 4 - випадний список розширень імен файлів

### 2.2.1.3. Друк текстового документа.

Якщо ви закінчили оформлення текстового документа, то, можливо, вам потрібно його роздрукувати. Для цього увійдіть у пункт меню **“Файл”**

Рис. 82. Група підпунктів меню **“Файл”** (**“File”**), які призначені для друку документа

Перш ніж роздруковувати документ, доцільно переглянути його розміщення на листках паперу. Якщо розміщення тексту на листках паперу вас влаштовує, переходьте до друкування документа. На рис. 83 подано діалогове вікно **“Друк”** процесора **Abiword**

Рис. 83. Діалогове вікно **“Друк”**

**“Принтер”** - закладка вікна вибору пристрою друку (у тім числі і вибір друку у pdf-файл), **“Аркуш”** - закладка вікна задання параметрів аркуша паперу і орієнтації тексту на ньому

Після задання необхідних параметрів (або прийняття параметрів **“за замовчуванням”**) вам залишається **„вказати мишкою”** на **“Друк”** і дочекатися завершення друку.

#### 2.2.1.4. Пошук заданої інформації

Завдяки пошуковій функції текстового процесора **Abiword** ви можете швидко знайти певний текст у текстовому документі. Для запуску пошуку вам потрібно у пункті меню **“Редагування”** вибрати пункт **“Знайти”**.

Рис. 84. Зміст пункту меню **“Редагування”**

У діалоговому вікні **“Знайти”** можна задати зразок пошуку, вікні **“Замінити”** – зразки пошуку і заміни. Вікно **“Перейти”** організовує навігацію по тексту.

а) б)  
Рис. 85. Вікна **“Знайти”** (а) і **“Замінити”** (б)

Рис. 86. Вікно **“Перейти”**

#### 2.2.1.5. Робота з таблицями

Табличну форму подання інформації широко використовують у ділових документах, для поєднання в одному документі тексту і таблиць можна скористатись з пункту меню **“Table”** текстового процесора **Abiword**

Рис. 87. Пункт меню **“Table”** текстового процесора **Abiword**

Після того, як ви **“виберете”** мишкою **“Table - Insert - Table”**, на екрані дисплея з’явиться діалогове вікно **”Вставити таблицю”** (**“Insert Table”**).



### Рис. 88. Вікно задання параметрів таблиці

Після заповнення полів цього вікна у документі з'явиться заготовка таблиці з заданою кількістю рядків і стовбців, яку можна заповнювати потрібною вам інформацією (слід наголосити на деякій нестабільності процесора **Abiword** при роботі з таблицями, тому під час заповнення таблиці виконуйте збереження набраної інформації частіше ніж звичайно).

#### 2.2.1.6. Робота з графікою

Текстовий процесор **Abiword** дозволяє вставляти у текстовий документ графічні зображення, для цього потрібно вибрати відповідний пункт у меню "**Вставити**"

### Рис. 89. Пункти меню "**Вставити - Зображення**"

Після вставки зображення його можна відмасштабувати за допомогою точок деформації (вісім сірих квадратиків) на зображенні і задати його розташування у тексті.

### Рис.90. Масштабування зображення

Слід наголосити, що увесь текст щодо роботи з процесором **Abiword** автор створив у цьому процесорі. Копії екранів були отримані за допомогою графічного редактора **Gimp** (див. далі).

#### 2.2.2. Табличний процесор **Gnumeric**

Після запуску табличного процесора **Gnumeric** на екрані дисплея відкриється вікно  
Рис. 91. Вікно табличного процесора **Gnumeric** з новою таблицею

Для уведення текстової або числової інформації (ціла частина від дробової відділяється комою! [для регіону "Україна"]) у комірку електронної таблиці потрібно двічі "клацнути лівою клавішею" мишки по комірці і ввести потрібну інформацію. Уведення формули розпочинається з уведення знаку =, далі можна або ввести формулу самостійно, або ж скористатись із майстра побудови функцій (див. далі).

- а)
- б)
- в)

Рис. 92. Приклади уведення інформації у комірки електронної таблиці (а – текстова інформація, б – числова, в – формульна **=sum(B2:AF2)**)

Для ілюстрації можливостей табличного процесора **Gnumeric** розглянемо задачу нарахування заробітної плати працівникам малого підприємства із погодинною оплатою праці. У цій задачі заповнимо 3 аркуші електронної книги (електронну таблицю із кількома аркушами називають електронною книгою):

- табелі обліку робочого часу за місяць із сумою відпрацьованих годин;
- відомість нарахування заробітної плати ;
- зведену таблицю розподілу коштів фонду оплати праці малого підприємства.

Після заповнення таблицю обліку робочого часу уведемо формулу підрахунку суми відпрацьованих годин. Для цього зробимо активною комірку **AG2** і “натиснемо мишко” знак  $\Sigma$  на панелі інструментів **Gnumeric**. У комірці **AG2** з'явиться текст **=sum()**. Задамо початкову (**B2**) і кінцеву (**AF2**) комірки сумування, розділені двокрапкою, і натиснемо клавішу **Enter** – у комірці з'явиться значення суми (див. рис. 91). Оскільки інформація в електронній таблиці може розташовуватись на кількох аркушах, то для використання числових значень з іншого аркуша використовують повне ім'я комірки - **ім'я\_аркуша!ім'я\_комірки**, наприклад: **табель!ag2** (число з комірки **ag2** на аркуші з іменем **табель**). Для зміни імені аркуша із стандартного **Аркуш1** можна використати функцію перейменувати контекстного меню – “вказіть мишкою” на ім'я аркуша, натисніть праву клавішу мишки і виберіть **“Rename”**. Після цього уведіть нове ім'я аркуша. При розв'язанні поставленої задачі окрім обчислення суми виникає завдання обчислень з умовою – внески у пенсійний фонд і фонд соціального страхування залежать від суми заробітку за місяць. Якщо працівнику нараховано за місяць менш 150 гривень, то він сплачує у пенсійний фонд 1% від нарахованої суми. Коли ж нарахована сума більша 150 гривень, то у пенсійний фонд вноситься 2% від нарахованої суми. Аналогічно у фонд соціального страхування перераховується 1% при нарахованій платні до 365 гривень, 2% при нарахованій платні більше 365 гривень. Для проведення таких обчислень використовують функцію **if**. На рис. 93 проілюстровано уведення цієї формули за допомогою гуру побудови функцій

Рис. 93. Вибір функції **if** із групи **“Logic”**

Рис. 94. Гуру побудови функції **ЯКЩО**

1 – поле логічного виразу (заробіток менше 150 гривень [заробіток обчислено у комірці b2]), 2 – формула для обчислень, якщо логічний вираз істинний, 3 – формула для обчислень, якщо логічний вираз хибний

Уведіть потрібні вирази у відповідні рядки. Введення кожного виразу завершуйте натисканням клавіші **Enter**. Після заповнення потрібних полів отримаємо наступну функцію

**=if (B2<150;B2/100;B2/50)**

Табличний процесор може будувати діаграми на основі заданого вами ряду чисел. На рис. 95 подана колова діаграма розподілу коштів оплати праці малого підприємства, а на рис. 96 – кроки побудови діаграми за допомогою гуру побудови діаграм

Рис 95. Колова діаграма розподілу коштів оплати праці

Виділіть лівою клавішею мишки ряд значень, на яких будуватиметься діаграма і виберіть у меню **“Вставка - Chart”** та “натисніть **”Вперед”** – на екрані дисплея відкриється вікно побудови діаграм

а)

б)

Рис. 96. Перший (а) і другий (б) кроки побудови діаграми

“Натисніть мишкою **Гаразд”** і вкажіть за допомогою лівої клавіші мишки місце розташування діаграми – на вказаному місці з'явиться вибрана вами діаграма. Якщо ви захочете перемістити діаграму в інше місце, клацніть по ній лівою клавішею мишки, наведіть курсор мишки на край рамки діаграми і, коли форма курсора перетвориться на чотиристоронню стрілку, затисніть ліву клавішу мишки та потягніть діаграму у потрібне вам

місце.

Для засвоєння викладеного матеріалу спробуйте повторити розв'язання задачі про нарахування заробітної плати працівникам із погодинною оплатою праці. При виконанні цієї задачі врахуйте наступні обставини (нормативи вказані на момент написання книги)

- із заробітної плати громадян України вираховуються внески у фонди:  
пенсійний (1% від заробітку при сумі до 150 гривень, 2% при сумі більше 150 гривень);  
соціального страхування (1% від заробітку при сумі до 365 гривень, 2% при сумі більше 365 гривень);  
боротьби із безробіттям – 0,5%.
- прибутковий податок обчислюється за ставкою 13% від різниці між нарахованим заробітком і внесками у фонди (пенсійний, соціального страхування і боротьби із безробіттям )
- окрім коштів, вирахованих із платні громадян, фонди отримують кошти із підприємств за нормативами:  
пенсійний – 32% від загальної суми нарахувань;  
соціального страхування - 4% від загальної суми нарахувань;  
боротьби із безробіттям - 1% від загальної суми нарахувань.

### 2.3. Опрацювання графічної інформації

Для опрацювання графічної інформації використовують графічні редактори. Створення і опрацювання зображень (графіків, діаграм, малюнків, креслень) є предметом діяльності великого розділу інформатики – комп'ютерної графіки. Зображення на екрані дисплею складається з величезної кількості кольорових крапок ( від 640\*480 до 2048\*1536). Різні кольори отримуються змішуванням трьох основних кольорів – синього (**B**), червоного (**R**), зеленого (**G**), тобто кольорова крапка на екрані дисплею насправді складається з трьох дуже близько розташованих синьої, червоної і зеленої крапок зі змінюваною інтенсивністю свічення. Графіка, яка ґрунтується на формуванні зображення як сукупності кольорових крапок, називається растровою (точковою). Кількість бітів, використовуваних для запам'ятовування окремої крапки, визначає кількість відтінків кольорів у зображенні:

- 1 біт – чорно-біле зображення (0 – чорна крапка, 1 – біла крапка);
- 4 біти – кольорове зображення з 16 кольорами;
- 16 біт – кольорове зображення з 65535 кольорів;
- 24 біти - кольорове зображення з 16700000 кольорами.

Оскільки на кожну крапку зображення доводиться відводити від половини до трьох байтів, то використання растрової графіки призводить до великих розмірів файлів, у яких зберігається графічне зображення. Програми для роботи з растровою графікою ми розглянемо пізніше. Окрім растрової графіки використовують векторну графіку, в якій зображення складають з окремих елементів – ліній, прямокутників, кіл та еліпсів, об'ємних фігур. Файли графічних зображень, побудованих у векторній графіці, менші по розміру від файлів растрової графіки. Графічні зображення, отримані у векторній графіці, добре масштабуються (збільшуються або зменшуються), причому, на відміну від растрової графіки, збільшення зображення відбувається без втрати чіткості зображення. Розглянемо графічні редактори, які можна використовувати в ОС **Linux** (редактор векторної графіки **OpenOffice.org.Draw** описано у п.2.1.3).

#### 2.3.1. Графічний редактор **KolourPaint**

Після запуску **KolourPaint** на екрані з'явиться вікно

Рис. 97. Графічний редактор **KolourPaint**

1 – меню програми, 2 – панель інструментів програми, 3 – інструменти для графічних побудов, 4 - поточний колір, 5 – палітра кольорів

Робота з редактором **KolourPaint** дуже подібна до малювання на аркуші паперу різноколірними олівцями і пензлями. Очевидно, що результати малювання залежать від вправності того, хто малює. На рис. 98 показано окремі елементи, з яких можна скласти зображення, працюючи у **KolourPaint**

Рис. 98. Графічні примітиви редактора **KolourPaint**

### 2.3.2. Графічний редактор **Gimp**

Графічний редактор **Gimp**, який за функціональністю співмірний з **Photoshop 6.0**, можна використовувати як в ОС **Linux**, так і у **Microsoft Windows**. Розглянемо короткий опис цього редактора. Після запуску програми на екрані дисплея з'являться вікна

Рис. 99. Вікна програми **Gimp**

Якщо навести вказівник "мишки" на інструмент **Gimp**, то під інструментом з'явиться пояснювальний напис про призначення інструмента.

Розглянемо структуру меню **Gimp**. У пункті “**Файл**” зосереджені операції “**Новий – Відкрити – Зберегти – Зберегти як – Відновити – Відіслати зображення – Друкувати – Закрити – Вийти**”.

Пункт “**Редагування**” містить дії по вирізанню, копіюванню, вставленню, відміні і повтору дій.

У пункті “**Виділення**” - різні виокремлення: всього зображення, кольорових областей, плаваюче і інверсне виділення, запис виділеного фрагмента містить дії по вирізанню, копіюванню, вставленню, відміні і повтору дій.

Пункт “**Вигляд**” містить дії для зміни масштабу зображення, показ/приховування лінійок і скеровуючих, відкриття нового вікна.

Основні маніпуляції з зображенням внесені у пункт “**Зображення**” - перетворення (в RGB, індексовані кольори, градації сірого), трансформації, обертання, обрізання.

Пункт “**Шари**” містить дії над шарами зображення.

В пункті “**Інструменти**” зосереджені основні інструменти для дій із зображенням (виділення областей різної форми, дзеркалювання, обертання і т.ін.)

Пункт “**Діалоги**” містить виклик додаткових панелей задання шарів, каналів, пензлів, палітр, шаблонів, градієнтів і т.ін.

Для накладання різноманітних спецефектів використовують пункт “**Фільтри**”. На рис. 100 подано результат роботи спецефекту “**Фільтри - Викривлення – Загнута сторінка**”, застосованого до рис. 99

Рис. 100. Спецефект “**Загнута сторінка**”

Пункт “**Відео**” містить дії з відеопослідовностями – розібрати відеопослідовність на кадри, змінити послідовність кадрів, видалити деякі кадри, зібрати відеопослідовність із окремих файлів.

Додаткову інформацію щодо роботи з графічним редактором **GIMP** можна знайти за адресою [www.gimp.linux.ru.net](http://www.gimp.linux.ru.net)

### 2.3.3. Редактор векторної графіки **InkScape**

Графічний редактор **InkScape** призначений для створення простої двовимірної векторної графіки і оперує форматом **SVG (Scalable Vector Graphics)**. Це відкритий формат, побудований на основі розширеної мови розмітки **XML**. Створювався цей формат з метою уможливити використання векторної графіки у оформленні веб-сторінок, де поки-що домінує ємка і незручна растрова графіка. На сьогодні це лідер серед вільного програмного забезпечення для роботи із графікою за функціональністю і якістю.

Перше, що слід запам'ятати – **Inkscape** працює із об'єктами. Об'єктом може бути будь-який елемент векторного зображення, наприклад крива, багатокутник, текст тощо. Кожен елемент означений і володіє властивостями, як от власні розміри, колір та товщина лінії та заповнення багатокутника, розмір і гарнітура шрифту. Також фіксується співвідношення між об'єктами та положення кожного із них. Криві у векторній графіці описуються за допомогою вузлів, від яких відходять спрямувальні риски (“вуса”), що “відтягують” у свій бік криву, тим більше, чим вони довші. Таким чином можна описати доволі складну криву за допомогою кількох вузлів із вусами, і легко модифікувати криву, пересуваючи вузли і вуса. Такі криві називаються кривими Безьє (Bezier), за іменем математика, котрий, власне, їх винайшов.

Рис. 101. 1 – меню, 2 – панель команд, 3 – панель параметрів, 4 – панель інструментів, 5 – рядок стану

Після запуску **Inkscape** створює новий документ у вигляді аркуша паперу. Редактор має нагорі головне меню, через нього можна отримати доступ практично до всіх дій над об'єктами і властивостей, за виключенням інструментів для створення нових об'єктів. Панель інструментів розміщена ліворуч, вертикально. На ній розміщені кнопки-піктограми, які викликають певні функції, у такому порядку: виділення і трансформації, редагування вузлів кривих, зміна масштабу, створення прямокутників, кіл, багатокутників, спіралей, рисування контурів, кривих, каліграфічних ліній, створення текстових об'єктів, ліній з'єднання та градієнтів кольору, а також засіб для вибирання кольору із вже наявних об'єктів. Наведення на інструмент подає коротку підказку. Нагорі під головним меню містяться панелі команд і параметрів. В першій (верхній) панелі команд можна отримати доступ до найпоширеніших команд, наприклад записати файл чи відмінити свої попередні дії, налаштувати сам редактор, виправити властивості контура, тощо. Панель параметрів змінюється відповідно до виділеного поточного об'єкта, подає його розміри, координати, особливості оформлення. Ці параметри можна виправляти прямо у полях даних. Внизу вікна програми видно рядок стану, який інформує про поточний об'єкт, та подає підказки до дій. Розглянемо методи роботи з **Inkscape** на прикладі прямокутника. Для цього на бічній панелі виберемо інструмент у вигляді синього прямокутника. Курсор змінить свою форму на маленький прямокутник. Натиснувши на аркуші на місці, де має бути верхній лівий кут і не відпускаючи мишки, тягнемо до потрібного розміру. В результаті отримуємо синій прямокутник. Якщо ми повернемо стан інструменту до вказівника, стрілку, то навколо об'єкта з'являються стрілки. Якщо за них потягнути, то можна змінювати розміри об'єкта. Якщо ще раз клацнути вказівником на ньому, то стрілки перетворяться на кругові, а також з'явиться центр обертання. Зачіпаючи за ці стрілки, ми можемо повертати об'єктом. Згодом можна також редагувати інші властивості цього об'єкта, за допомогою підпунктів меню “Об'єкт” в головному меню. Найбільш потрібні властивості **Заповнення і штрих**, де можна поміняти колір заповнення, колір штриха та його товщину.

Рис. 102. Властивості заповнення і штриха для об'єкта.

Працюючи в **Inkscape** можна накладати кілька фігур одна на одну, міняти порядок, відсуваючи деякі на тло, щоб не затуляли головну фігуру. Залежно від прозорості заповнення переднього об'єкта буде видно також і задній. В одному рисунку можна створювати кілька шарів, що потім зручно було ними маніпулювати, в рисунок можна вставити растровий рисунок і малювати поверх нього.

Розглянемо структуру нашого прямокутника з точки зору кривих Безьє. Для цього виберемо відповідний інструмент для виправлення вузлів. На прямокутнику ми побачимо два квадратики (це власне вузли), і один кружечок (це закінчення вуса). Перетягуючи вузли, міняємо розміри прямокутника, а перетягуючи закінчення вуса, заокруглюємо кути прямокутника.

Звичайно, створення векторної графіки – доволі складний і клопітний процес, і щоб нарисувати складний реалістичний малюнок, треба витратити багато часу. Отриманий результат можна записати у форматі **svg**, які мають читати інші сучасні редактори векторної графіки, а також у форматах **eps** та **ai**, що популярні у програмах комп'ютерної верстки. Слід пам'ятати, що повністю можливості для подальшого редагування зберігаються лише у форматі **svg**. Зображення також можна експортувати в растровий формат **png**, задаючи потрібні розміри і роздільну здатність растру.

У пункті меню “Довідка” розташовані виклики доволі докладних підручників (англомовних) з описом методів роботи з редактором **Inkscape**

Рис. 103. Підручники з описом методів роботи з редактором **Inkscape**

#### 2.4. Машинний переклад тексту

Здатність ПЕОМ швидко шукати зразки тексту у текстовому документі (див. “Текстовий процесор...”) є основою систем машинного перекладу текстів. Такі програми розроблялися для великих ЕОМ ще до появи персональних ЕОМ. Масове поширення ПЕОМ спровокувало створення програм машинного перекладу для них. Сучасні програми машинного перекладу можна розділити на два класи програм:

машинні словники;

системи машинного перекладу текстів.

За понад 20-річну історію існування ПЕОМ машинних словників для них було створено дуже багато (на початку 90-х років ХХ-сторіччя у Львові для **MS DOS** були створені такі програми – оболонка двомовного словника **Vocab** [ЛНУ імені Івана Франка], спеціалізована база даних “Слово” [Львівський медичний інститут]. На жаль, ці роботи не отримали подальшого розвитку). Незважаючи на велику кількість машинних словників робота із ними дуже подібна – вибираєте потрібний словник і напрямок перекладу, у рядку задання слова набираєте слово, для якого потрібно знайти переклад, і натискаєте клавішу **Enter**. У вікні перекладу отримуєте певну кількість варіантів перекладу (рис. 104 – 106)

Рис. 104. Вікно програми **Word Translator** з розкритим пунктом меню **Dictionaries**

а)

б)

Рис.105. Приклади роботи програми **Word Translator** ( а) англо-німецький переклад [переклад слова **home**], б) – німецько- англійський переклад [переклад слова **Wiederschen**])

а)

б)

Рис.106. Приклади роботи програми **Stardict**

а – переклад заданого слова, б – вибір перекладу слова із слів, які починаються із літери d

Машинний словник **Stardict** можна використовувати як в ОС **Linux**, так і в ОС **Microsoft Windows**. Він дуже простий, надає змогу сканувати виділення на льоту, інтегрується в різні графічні середовища. Для автоматичних словників використовують словникові бази в різних форматах. Але найпоширеніші із них – це формати **dict** та **mova**. Цей формат використовують для онлайн-словників, наприклад [www.dict.org](http://www.dict.org). Із цих форматів словники можна конвертувати у формат **stardict**. Зусиллями спільноти на тепер є непоганий вибір словникових баз, які можна звантажити наприклад із [http://stardict.sourceforge.net/Dictionaries\\_dictd-www.mova.org.php](http://stardict.sourceforge.net/Dictionaries_dictd-www.mova.org.php) у вигляді архівного файлу і розархівувати його в каталог `/usr/share/stardict/dic`. За наступним завантаженням **Stardict** зареєструє появу нових словників і додасть їх у список наявних на ПЕОМ словників. Міняти почерговість словників і вимикати декотрі із них можна за допомогою конфігурування самого **Stardict**.

## 2.5. Програмування в ОС **Linux**

### 2.5.1. Історичний спадок

У старі часи саму операційну систему **Unix** нерідко називали системою програмування. Глибокий зв'язок систем цієї родини із програмуванням заклав підстави для її активного розвитку. Орієнтація цієї системи на професіоналів сприяла формуванню особливого стилю програмування. Сама мова програмування **Cі** була створена під час перенесення **Unix** з міні-ЕОМ **PDP 8** на **PDP 11**.

Цей розділ присвячений загальному огляду засобів та інструментарію для побудови програм, які пропонують нам до використання сучасні збірки ОС **Linux**. Самого процесу програмування деякою певною мовою ми не торкатимемося, оскільки це вимагало б окремої книжки.

### 2.5.2. Засоби

Для того, щоб скласти програму і виконати за її допомогою певну дію, нам потрібно мати у розпорядженні такі основні компоненти:

- текстовий редактор для створення тексту програми;
- компілятор для створення виконавчого коду (або інтерпретатор для безпосереднього виконання);

- відладник для довершення і видалення помилок;
- засоби для використання бібліотек та самі бібліотеки.

Редактор та відладник разом із системою підручної документації часто укладають в інтегроване середовище розробника для забезпечення зручності роботи програміста. В **Linux** основу системи розробки складає так званий **GNU toolchain**, у переклад набір інструментів.

Цей набір складається із:

**GNU Compiler Collection (gcc)** – комплекту компіляторів цілого ряду мов програмування, сконструйованих на спільній базі. Це у першу чергу компілятор мови **Cі** і **Cі++**, а також **Objective C**, **Java**, **Ada**, **Fortran**. Ймовірно, це найбільш універсальний із всіх коли-небудь створених інструментів розробки, адже версії цього компілятора портовані на всі можливі обчислювальні платформи. Компілятор має розвинутий набір опцій для командного інтерфейса, докладне вивірення помилок та строго дотримується опублікованих міжнародних стандартів. Компілятор супроводжується набором базових бібліотек **GNU C library (glibc)**. Останні версії цього компілятора наділені здатністю продукувати швидкісний виконавчий код, оптимізований на виконання процесорами нових поколінь;

**GNU binutils** – пакет, який включає асемблер, компоувальник (**linker**), інші базові засоби;

**GNU debugger (gdb)** – відладник, який надає всі необхідні для відлагодження засоби (наприклад покрокове виконання, вхід у функції, прогляд умісту стеку, значень та адрес змінних). Цей відладник підтримує ряд форматів виконавчого коду, механізми дослідження пам'яті, тощо. Відладник побудований на основі командного інтерфейсу, але до нього легко добудувати графічну оболонку;

**GNU make** – засобу автоматизації роботи із великими проектами, а також засоби конфігурування цієї системи, наприклад **autoconf**.

**GNU toolchain** доповнюється стандартними бібліотеками мови **C**, **GNU libc**, а також **libtool**, засобами для створення і керування динамічними і статичними бібліотеками.

По ходу створення якоїсь конкретної програми у систему встановлюються додаткові бібліотеки (розробницькі версії), додаткові інструменти. Окрім **GCC**, у **Linux** можна використовувати окремі компілятори. Особливо актуальним для здійснення процесу навчання є компілятор мови програмування **PASCAL**. Існує його вільна реалізація **FreePascal** у вигляді компілятора, реалізованого для різних платформ. Цей компілятор підтримує стандарти на мову програмування **PASCAL**, а також всі базові розширення **TurboPascal 7.0** та **DELPHI 2.0**.

Із сайту компанії **Borland** можна звантажити безоплатну версію інтегрованого середовища розробки **Kylix 3.0**, яке цілком відтворює зовнішній вигляд і практично сумісне з середовищем **Delphi 7.0**. Безоплатна версія (**OpenEdition**) дещо обмежена в можливостях, особливо у тому що стосується баз даних, але для більшості звичайних задач є цілком достатньою.

### 2.5.3. Бібліотеки

Пересічна система розробника на базі ОС **Linux** містить близько кількох сотень динамічних бібліотек. Ці бібліотеки переважно орієнтовані на мови програмування родини **Cі**, але їх також можна викликати із інших мов програмування за допомогою системи прив'язок. Слід звернути увагу на те, що, крім самих бібліотек, які необхідні для виконання укомпонованих з ними прикладних програм, можна доставити пакунки для розробки (позначені як **dev**). За допомогою цих заголовочних файлів можна використовувати ці бібліотеки для розробки. Більшість бібліотек мають добре документованій програмний інтерфейс. Загалом бібліотеки можна класифікувати за призначенням, наприклад забезпечення операцій із файлами графічних зображень, доступ до баз даних, забезпечення побудови графічного інтерфейсу, ефективні бібліотеки для числових обчислень, маніпуляцій тривимірними об'єктами, мультимедійних операцій. Слід відмітити, що у сфері розробки графічного інтерфейсу користувача конкурують дві основні гілки. Одна із них – набір засобів розробки для мови програмування **Cі++** на основі бібліотеки графічних елементів інтерфейсу (віджетів) **QT** та доповнювальних бібліотек **KDE**, які дають змогу розробляти програми для інтегрування у графічне середовище **KDE**. Інша гілка – бібліотеки графічних елементів **Gtk+** та елементи середовища **GNOME**, які орієнтовані на мову програмування **Cі**. Саме ці бібліотеки переважно формують зовнішній вигляд графічного інтерфейса користувача **Linux**, хоча існують також інші графічні бібліотеки. Це надзвичайне багатство надає розробнику широкі можливості для втілення різноманітних задумів.

### 2.5.4. Утиліти

Щоб ефективно програмувати, в ОС **Linux** застосовують ряд потужних супровідних засобів для полегшення роботи з комплексом файлів з вихідними текстами програм, пошуку помилок, узгодження варіантів розробки, а також універсальні засоби для опублікування розробок в глобальній мережі, створення документації. Ці функції виконує частина **GNU**



**toolchain** – система керування проектами, **make**. Ця відносно проста і компактна система дає змогу керувати проектами, що складаються із величезної кількості окремих файлів і бібліотек, за допомогою одного файлу. Ця утиліта доповнює спеціальний набір макрокоманд **autoconf** та **automake** для генерування файлу **makefile** відповідно до конкретних потреб платформи чи налаштувань. Разом вони складають ту основу стандартизації проектів вільного програмного забезпечення, яка забезпечує універсальність і перенесення ПЗ між різними платформами. Можливо, для новачка у світі **Linux** і буде незвичним такий спосіб укладання проектів, тоді він може скористатися засобами, які надають змогу генерувати необхідні файли, не знаючи самих макрокоманд чи мови **makefile**.

Важливою частиною, яка потрібна для здійснення колективної розробки, є система контролю поточної версії, яка складається з серверної сторони (сервер **CVS**), а також з допоміжних програм клієнта (наприклад, клієнти служби **CVS**, утиліта **patch**).

Всі ці засоби разом із низкою інших дають широкі можливості для реалізації проектів довільного масштабу, а також навчають принципам колективної роботи над програмами, чіткого і виразного стилю програмування. Також це надає можливість вільного перевикористання вже готових наробок інших програмістів (ліцензування більшості опрограмування для **Linux** таке дозволяє), що пришвидшує розробку нових програм.

### 2.5.5. Інтегровані системи розробки

Від попередників зі світу **Unix** у спадок **Linux** отримав ряд “класичних” інтегрованих середовищ на базі розвинутих текстових редакторів. Це, наприклад, системи **Emacs** (разом із його похідними), редактор **Vim**, система **Eclipse**. Завдяки своїй потужності і ефективності вони беззаперечно утримують свою популярність серед фахівців, але для навчання необхідні простіші засоби, доступні для освоєння і новачку, зовнішньо подібні на аналогічні інструменти на платформі **Microsoft Windows**. У відповідь на потреби програмістів-новачків **Linux** за останні роки збагатився новими графічними оболонками розробника, які пропонують зручне середовище для розробника (**Kdevelop**, **Anjuta** тощо). Ці системи у зручному вигляді інтегрують у собі разом практично всі вказані у попередніх параграфах цього розділу засоби. Вони подають інструментарій для інтерактивного створення проектів, а також кожне із них підтримує цілу низку мов програмування.

У зв'язку з популярністю візуального програмування до цих систем докладають засоби швидкої розробки аплікацій для різних систем графічних бібліотек. Це **QTDesigner** для розробки на **QT**, та **Glade** для розробки на основі **Gtk+**. За необхідності можна також скористатися інтегрованими середовищами для консолі, наприклад **xwpe**, **rhide**. За принципами роботи вони схожі на оболонку **TurboPascal**.

Таким чином, на основі всього сказаного можна стверджувати, що потрібний інструментарій розробки присутній в необхідній кількості і якості для програмування як професійного, так і аматорського; технічна і загальна документація до цих засобів наявна (щоправда, переважно англійською мовою); свобода цих засобів забезпечує не лише значну економію коштів, а й можливість доробки їх власними силами; вільне програмування надає доступ до широкого спектру зразків.

### 2.5.6. Інтегроване середовище розробки **Anjuta**

Хоча оболонка програміста **Anjuta** підтримує велику кількість мов програмування, основне її призначення – мови **Cі** та **Cі++**. Оберемо її для прикладу програмування мовою **Cі** та пояснимо основні елементи.

Рис. 107. Вікно програми **Anujta**

Запустивши **Anjuta IDE**, можна згенерувати за допомогою помічника новий проект. Для цього слід зайти у меню **Файл** та обрати пункт **Новий проект**. Зробіть крок, і перед вами постане питання вибору типу проекту – чи то буде програма мовою **Сі**, що працюватиме у текстовому режимі, або програма із уживанням набору елементів графічного інтерфейса **Gtk+**, чи ви будете використовувати також частини графічного середовища **GNOME**. На наступному кроці оберіть мову програмування (**Сі**, **Сі++** чи змішаний) проект, дайте проектіві назву і присвойте початкову версію. Далі надайте проектіві короткий опис і будьте не лише короткі у формулюванні, але й точні. Чим чіткіше уявлятимете, що саме має робити програма, тим більші ваші шанси на успіх.

Якщо ви маєте намір поширювати свою програму на засадах **GNU**, то у наступному вікні зазначте це, та увімкніть підтримку **gettext**, адже це надасть змогу легко інтернаціоналізувати ваше творіння.

Після створення тексту програми його доцільно зберегти на магнітному диску. Дала переходимо до компіляції програми (**Скласти - Скомпілювати** [або **F9**]). Якщо компіляція завершилась успішно, то програму можна виконати (**Скласти - Виконати** [або **F3**]).

У пункті меню **Довідка** оболонки **Anujta** ви знайдете рубрики **Anujta Tutorial**, **Anujta User Manual**, **Anujta FAQ List** з інформацією про користування оболонкою.

#### 2.5.7. Оболонка **Free Pascal** для програмування мовою Паскаль

Після запуску оболонки **Free Pascal** на екрані дисплея відкриється вікно

Рис. 108. Вікно оболонки **Free Pascal**

Для створення нової програми виберіть у меню **File – New**, відкриття записаної на ЖМД програми **File – Open**. Після завершення роботи над текстом програми виберіть у меню **Compile**. Якщо компілятор виявив у тексті програми помилки – виправте їх. Після отримання повідомлення компілятора про успішне завершення процесу компіляції складену програму можна запустити на виконання.

## 2.5.8. Оболонка **Algo** для програмування мовою Паскаль

Оболонка **Algo** створена для операційної системи **Microsoft Windows** і може виконуватись за допомогою системи **Wine** в операційній системі **Linux**. Після запуску оболонки на екрані дисплея з'явиться вікно

Рис.109. Вікно оболонки **Algo**

1– меню програми, 2 – панель інструментів, 3 – вікно тексту програми, 4 – лінійка прокручування вікна тексту програми у вертикальному напрямку, 5 – вікно результатів роботи програми, 6 – лінійка прокручування вікна результатів у вертикальному напрямку, 7-8 - лінійки прокручування вікна тексту програми і вікна результатів у горизонтальному напрямку, 9 – рядок стану програми

Текст програми можна набирати з клавіатури повністю, або ж доповнювати текст, складений із шаблонів (рис.110), символами введеними з клавіатури.

Рис. 110. Шаплони для вибору службових слів і команд Паскалю

Шаплони подаються українською або англійською мовами залежно від встановленого варіанту (**Ukr** – українською мовою, **En** – англійською мовою). Службові слова Паскалю відображаються синім кольором, типи змінних і імена стандартних функцій – червоним, коментарі – зеленим, числа та імена змінних – чорним. На рис. 111 подано вікно оболонки **Algo** з програмою побудови графіка функції з україномовним написанням службових слів Паскалю.

Рис.111. Вікно оболонки **Algo** з програмою побудови графіка функції

Призначення окремих інструментів оболонки можна взнати, якщо навести на них вказівник “мишки”. Разом з тим розглянемо деякі інструменти оболонки:

- впорядкувати текст програми (як на рис. 111);
- виконати програму;
- виконати один крок програми (команда, яка буде виконуватись наступною, підсвічується зеленим кольором);
- писати шаплони українською мовою;
- писати шаплони англійською мовою.

В україномовній довідковій системі оболонки **Algo** (пункт меню “Довідка”) подана інформація про користування оболонкою, мову програмування Паскаль і додаткові засоби оболонки (розширення Паскалю). Докладний опис оболонки **Algo** і зразки програм для неї подані у [3].

## 2.6. Педагогічні програмні засоби в ОС Linux

До складу оболонки **KDE 3.5** входить пакет навчальних програм, який встановлюється разом з оболонкою **KDE** (при потребі його можна встановити з пакунка **kdeedu-3.5**). На рис. 112 подано стартове меню **KDE** з групою програм “**Освітні**”

Рис. 112. Освітні програми оболонки **KDE**

До складу пакету навчальних програм входять програми, пов'язані з вивченням іноземних мов, математики, хімії, астрономії, інформатики та оболонка для проведення тестувань за вибраною темою з будь-якого предмета. Навчальні програми дозволяють вчителю самостійно формувати тип, мету та зміст уроку, вибирати темп роботи учня за ПЕОМ, а також надають можливість діяти за розробленим алгоритмом навчальної програми. Ці програми є дуже простими у використанні, володіють сучасним і зручним інструментарієм та інтерфейсом.

Розглянемо перший розділ навчальних програм, який називається **Мови**. До його складу входять програми:

**KHangMan**, яка дозволяє вивчати англійські слова. Форма подання завдань здійснюється у ігровій формі (див. рис. 113). Користувач може вибрати рівень (легкий, середній або складний) і тему. Програма також може змінювати палітру екрану, що значно поживляє навчальний процес.

Рис. 113. Гра **KHangMan**.

Зміст гри полягає у відгадуванні англійських літер у слові, яке вибирається випадково з власної бібліотеки програми. Той хто грає, бачить на екрані лише позиції літер у вигляді рисочок. При вдалій спробі, тобто вгаданій літері, з'являється відгадана літера. Кількість спроб є обмеженою;

**Kiten** – це програма, яка служить для вивчення японської мови;

**KLettres** – дозволяє в ігровій формі вивчати алфавіт англійської, французької, німецької або датської мов

Рис.114. Гра **KLettres**

Програма має чотири рівні гри, перемикач мов, а також вибір палітри екрану. Ідея гри полягає у повторенні літер, що з'являються на екрані, при відповідному натиску потрібної клавіші на клавіатурі. Складність гри поступово зростає, при цьому збільшується кількість літер, що з'являються на екрані. На відповідному етапі користувачу необхідно вводити літери по пам'яті, а також сприймаючи їх на слух;

**Kanagram** – це програма, яка у ігровій формі дозволяє вивчати англійські слова;

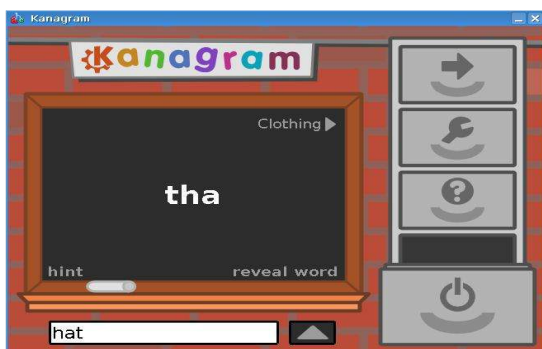


Рис. 115. Гра **Kanagram**

Ідея гри полягає у відгадуванні слова, яке зображене на екрані, але має довільно переставлені літери. Для того, щоб перемогти, користувачу необхідно ввести відповідне правильне слово. Кількість спроб є необмеженою. Існує можливість вибирати тему, створювати свій варіант словника, редагувати вибране та поновлювати тематичну базу з мережі

Інтернет. У програмі також передбачено вибір рівня складності, включення підказки, налаштування режиму роботи інтерфейсу, а також облік кількості спроб та формування списку переможців;

**Kverbos** служить для вивчення відмінювання правильних та неправильних дієслів іспанської мови. Користувач може сам формувати завдання, наповнювати словник дієсловами з перекладом їх на різні іноземні мови, редагувати дані;

**KVocTrain** - це програма, яка допомагає тренувати словниковий запас слів, коли ви вивчаєте іноземну мову. Вона надає можливість створювати і редагувати свій власний словник іноземних слів.

Ідея роботи з програмою - імітація флеш-карток. З одного боку поверхні картки написано слово іноземною мовою, з іншого подано його переклад. Вчитель може вибрати тренувальний режим навчання або режим тестування.

Другий розділ навчальних програм **Математика** складається з п'яти програм:

**Kig (Інтерактивна геометрія)** використовується для вивчення геометрії. Програма складається з трьох розділів - *конструювання, перетягування і вимірювання*.

Рис. 116. Програма **Kig** у режимі конструктора

*Конструктор* дозволяє будувати та досліджувати властивості простих геометричних об'єктів на декартовій площині. Він має зручне і просте у користуванні інструментальне меню, яке дозволяє побудувати точку, лінію, промінь, відрізок, вектор, трикутник, коло, сегмент тощо. Всі інструменти організовані в окремі групи, які легко, при потребі, можна переміщувати в потрібне місце екрану.

*Перетягування* геометричних фігур по площині рисунка дозволяє змінювати їх розміри та здійснювати топологічні перетворення (див. рис. 117).

Вибрана точка при переміщенні може залишати, при потребі, слід трасування. Якщо повторно звернутися до цієї точки, то властивість малювання траєкторії руху зберігається за нею. Під час переміщення точки режим трасування можна відключати і спостерігати лише самий процес зміни контурів геометричної фігури. Відповідний пункт меню панелі інструментів надає користувачу можливість змінювати колір і розміри фігури.

Рис. 117. **Kig** у режимі перетягування

*Вимірювання* дозволяє визначати відстань між двома точками, довжину кола, площу круга, зміщення між точками та величину кута (див. рис.118). При перетягуванні фігури і відповідній зміні її лінійних розмірів, величини, що характеризують ці параметри також змінюються. Ця властивість програми дозволяє формувати завдання на дослідження властивостей геометричних фігур або проводити відповідне "унаочнене" доведення геометричних теорем;

Рис. 118. **Kig** у режимі вимірювання

**KmPlot** є програмою для побудови графіків функцій в декартовій і полярній системах координат. Вона містить потужній інструментарій для налаштування кроку табуляції функції, кольорів та товщини ліній графіків, виставлення масштабу, тощо. Програма дозволяє задавати параметричні функції та виводити їх одночасно на екран;

Рис. 119. Побудова графіків у **KmPlot**

**KPercentage** є маленькою за розмірами і можливостями програмою, що допомагає учням розвинути їх здібності та уміння в обчисленні відсотків (процентів) від числа.

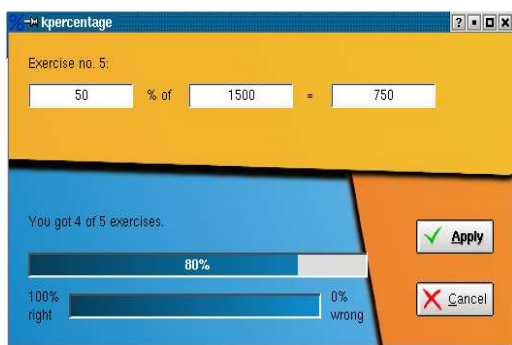


Рис.120. Обчислення в **Kpercentage**

Програма містить спеціальні тренувальні розділи для трьох базових задач на проценти із різними рівнями складності. Врешті решт, учень може самостійно вибирати "випадковий" рівень, тобто рівень при якому та чи інша задача генерується за принципом

випадкових чисел. Програма містить кнопки перемикавання рівнів складності та допомоги до відповідних алгоритмів обчислень.

**KBruch** є навчальною програмою, яка дозволяє в тренувальному режимі вивчати дії з дробами.

Програма містить чотири тематичних розділи. Перший розділ пов'язаний з основними арифметичними діями над дробами (додавання, віднімання, множення та ділення). Другий розділ дозволяє проводити порівняння дробів між собою з відповідним аналізом можливих помилок. Третій розділ – перетворення звичайних дробів у десяткові. Четвертий розділ розкриває тему розклад числа на множники. У програмі є можливість налаштувати задання типу дробу, максимальний головний знаменник та дії над дробами. Проводиться діагностика на кількість правильних і неправильних відповідей з відповідним аналізом помилок.

Рис.121. Дії з дробами в **Kbruch**

**LabPlot** – програма побудови графіків, діаграм, гістограм, а також обчислення диференціалів та інтегралів заданих користувачем функцій.

Третій розділ **Наука** містить всього дві програми з хімії та астрономії. Програма **Kalzium** є електронним варіантом таблиці Менделєєва.

Працюючи з нею, учень може одержувати багато різноманітної інформації про той чи інший хімічний елемент. Якщо даних недостатньо, то надається можливість підключитися через мережу Інтернет до відповідних інформаційних джерел. Програма дозволяє аналізувати таблицю хімічних елементів залежно від стану речовини - твердого, рідкого та газоподібного.

Рис. 122. Основне вікно програми **Kalzium**

У режимі симуляції, змінюючи температуру, можна бачити які речовин за даних умов переходять з одного агрегатного стану в інший. Додаткові вкладки програми дозволяють бачити зображення вибраного хімічного елемента (див. рис.123), атомарну модель рівнів електронів, хімічні довідникові дані, енергетичні характеристики та спектр.

Вмонтований калькулятор швидко і легко може обчислювати молярну масу речовини за введеною формулою хімічного елемента. Програма дозволяє досліджувати історію відкриття елементів за часовим фактором, тобто при виставленні відповідного року на спеціальній шкалі, в таблиці з'являються символи відомих на той час елементів.

Рис.123. Хімічний елемент уран

**Kalzium** містить також тести для визначення рівня знання учнем таблиці Менделєєва. Кількість питань можна задавати самостійно.

**Stars** - це небесна сфера зоряного неба з центром у точці спостереження. Карта зоряного неба містить близько 40 000 зірок різної зоряної величини та яскравості, 13 000 небесних об'єктів, усі планети Сонячної системи, Сонце та Місяць, Чумацький шлях, тощо. Існує можливість поетапного включення режиму основних елементів небесної сфери (точок і ліній), координатної сітки, екліптики, зодіакального поясу. Зорі відображаються на екрані згідно своїх природних кольорів і зоряних величин. Найяскравіші зорі мають підписану мітку (напр. Betelgeuse). По кутах зоряної карти розміщується інформація про поточний час (напр. "LT: 15:46:12 07/06/03), місце спостереження (напр. Yalta, Ukraine) і про текучий об'єкт в центрі екрану дисплея (напр. Focused on: nothing), а також небесні координати. Зверху екрану зоряного неба розміщуються панелі "**Головного меню**" та "**Перегляду**".

Рис.124. Програма **KStars**

Використовуючи можливості першої панелі, можна роздруковувати карту зоряного неба на пристрої друку, виставляти час та місце спостереження, вмикати режим руху небесних світил (від 0,1 сек до 100 років), змінювати масштаб спостереження, керувати конфігурацією виводу об'єктів на екран (тільки планети чи тільки зорі), тощо. Друга панель полегшує роботу із самою зоряною картою, дозволяючи вмикати лінії обрисів сузір'я, відображати дуже віддалені об'єкти, показувати Чумацький шлях, керувати режимом увімкнення/вимкнення системи небесних координат, підпису найважливіших об'єктів.

При наведенні і клацанні правою кнопкою мишки по об'єкту можна одержати вичерпну характеристику про нього. Утримуючи ліву кнопку мишки і перетягуючи її, можна повертати небесну сферу на свій розсуд.

Четвертий розділ освітніх програм **KDE 3.5** містить **Засоби для навчання**, який складається з програми **Keduca** (див. рис.123).

**KEduca** є тестовою оболонкою, яка дозволяє проводити поурочні тестування учнів за заданою темою, а також допомагає вчителю самостійно формувати завдання за допомогою комп'ютерного набору. Тест можна роздруковувати або використовувати у початковому електронному варіанті.

Рис.125. Програма **Keduca**

Завдання містить запитання і декілька варіантів відповідей. Тестуваний повинен прочитати питання і вибрати правильну відповідь на нього із загального списку. Пізніше підвести стрілку мишки до перемикача біля правильної відповіді і натиснути ліву кнопку. На кожне запитання відведено певний час, який демонструється відповідним показником (синя смужка з процентним індикатором). По завершенню відведеного часу, автоматично

завантажується наступне запитання. Учень може "економити" час відповіді, натискаючи кнопку *Next* і переходити самостійно до наступного запитання.

Тестова оболонка **KEduca** містить достатньо простий редактор для введення запитань і відповідей. Вибравши опцію головного меню **Build – Create - Modify...**, можна створювати новий тест або модифікувати вже існуючий. При формуванні завдання учитель може самостійно виставляти час, відведений на відповідь учня на певне запитання, вводити коефіцієнт складності, кількість відповідей, формувати підказку, відповідний текст пояснення, а також створювати і налаштовувати післятестове вікно аналізу результатів. Кількість варіантів відповідей є необмеженою.

У розділ **Різне** навчальних програм **KDE 3.5** входить традиційний клавіатурний тренажер.

Режими налаштування програми дозволяють підсвічувати наперед літеру для натиску на клавіатурі, визначати швидкість введення символів, виставляти рівень складності тексту. Програма має можливість перемикавання з режиму уведення літер на режим уведення цифр. Пункт меню **Опції/Головні** дозволяє налаштувати кольори відповідних клавіш, включити звуковий супровід та вибрати відповідний шрифт для зручності роботи.

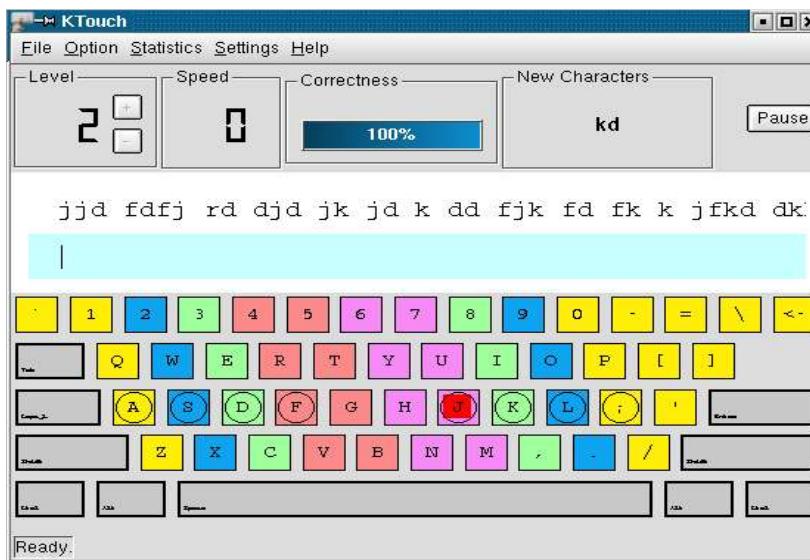


Рис. 126. Клавіатурний тренажер

## 2.7. Глобальна мережа Інтернет і засоби для роботи у ній

Інтернет – це всесвітня комп'ютерна мережа, яка зв'язує мільйони ЕОМ у всьому світі. Для функціонування і збереження цілісності такої інфраструктури кожний її учасник (вузол) дотримується певних визначених правил. Для функціонування такої складної мережі розроблено спеціальну мову, за допомогою якої спілкуються вузли мережі, а саме протоколи **ТСР/ІР**. Ці протоколи використовують для встановлення з'єднання між вузлами мережі і контролю передавання інформації між ними. Координати вузла задаються **ІР**-адресою, що займає 32 -біта, і описується чотирма цілими цифрами, наприклад:

219.162.201.204

Оскільки людям важко оперувати ланцюжками цілих чисел, тому замість **ІР**-адрес використовують доменні імена (буквально володіння). Таке ім'я складається з назв,



розділених крапками

ім'я ПЕОМ ...домен II рівня.домен I рівня

Домен I рівня визначає країну або тип організації, якій належить ПЕОМ, наприклад:

**ua** – Україна;

**ru** – Росія;

**us** – США;

**fr** – Франція;

**uk** – Великобританія;

**edu** - навчальні заклади;

**gov** - урядові заклади;

**net** - провайдери Інтернет;

**com** - комерційні організації.

Домен II рівня задає ім'я організації, яка володіє або керує мережею, до якої під'єднана ПЕОМ. Ім'я ПЕОМ визначає конкретну ПЕОМ у мережі, визначеній доменами I, II (і, можливо, наступних рівнів).

Розглянемо приклад доменної адреси:

Пов'язування доменного імені та **IP**-адреси здійснюється за допомогою спеціальної бази даних, яка розподілена між системою серверів імен (DNS).

Інтернет слугує з'єднувальним місточком між двома видами ЕОМ:

серверів, які надають послуги;

клієнтів, які доправляють до користувача інформацію і показують її відповідним чином.

### 2.7.1. Обладнання для під'єднання до глобальних мереж

Для під'єднання клієнтів до Інтернету можна використовувати цілу гаму пристроїв з доволі різними характеристиками. Донедавна основним видом під'єднання ПЕОМ до Інтернету було під'єднання по телефонних лініях з використанням модема (внутрішнього або зовнішнього)

Рис.127. Модеми

1 – внутрішній для стаціонарної ПЕОМ, 2 – внутрішній для переносної ПЕОМ, 3 - зовнішній

Максимальна швидкість передавання інформації, яку забезпечують модеми, складає 56 Кбіт/сек. При роботі з модемом телефонна лінія стає недоступною для телефонних розмов. Більш високу швидкість передавання інформації забезпечують так звані **xDSL**-модеми (якщо на вашій АТС встановлене цифрове обладнання). На рис. 128 зображена структура **xDSL**-мережі

Рис. 128. Структура **xDSL**-мережі

**xDSL**-з'єднання забезпечує швидкість передавання інформації до 52 Мбіт/сек. (залежно від типу модема і відстані до АТС). За рахунок розділення частот сигналів на **xDSL**-лінії одночасно можна розмовляти по телефону і працювати у Інтернеті.

Якщо до будівлі, у якій розташована ПЕОМ, не прокладена телефонна лінія, то під'єднатись до Інтернету можна через радіоінтерфеси, оптичні лінії зв'язку (дороге задоволення) або через супутник.

Найбільш поширеним в Україні бездротовим інтерфейсом є інтерфейс **IEEE 802.11.x**. Хоча цей інтерфейс розроблявся для бездротового під'єднання до локальних мереж на відстані до 300 м. на відкритій місцевості (у будівлях максимальна відстань зв'язку залежить від матеріалу, з якого виготовлені перегородки будівлі), в Україні його використовують для зв'язку на великі відстані завдяки використанню скерованих антен (фірма Уарнет підтримує такий зв'язок між Львовом і Сколе). Різні варіанти цього інтерфейсу забезпечують різну швидкість передавання інформації:

**IEEE 802.11.a** – 6-24Мбіт/сек.;

**IEEE 802.11.b** – 1-11Мбіт/сек.;

**IEEE 802.11.g** – до 54Мбіт/сек.

На рис. 129 -130 подано зображення пристроїв інтерфейсу **IEEE 802.11**

Рис. 129. Обладнання бездротового доступу стандарту **IEEE 802.11**

1 – контролер **IEEE 802.11** у форматі PCMCIA,

2 – PCI-контролер, 3 – антенне з коловою діаграмою, 4 – точка доступу

Рис. 130. Скерована антена стандарту **IEEE 802.11** з інтерфейсом **USB**

Минулого року в Україні абонетська плата за супутниковий доступ до Інтернету досягла прийняттого рівня 60 у.о. На рис. 131 подана структура симетричного каналу **DirecPC**. Окрім симетричного каналу можна використовувати асиметричний канал (запит на інформацію послається через модем з невисокою швидкістю, інформація до користувача з високою швидкістю надходить через супутник).

Рис. 131. Схема симетричного з'єднання ПЕОМ з використанням **DirecPC**

Ще одним методом бездротового доступу до Інтернету є використання мобільних телефонів із стандартом **GPRS** (General Packet Radio Service). Сьогодні в Україні **GPRS**-з'єднання здійснюють два провайдери – Київстар і UMC. Для з'єднання з ПЕОМ можна використати інтерфейси **RS-232**, **USB**, **Bluetooth**, **IrDA**. Максимальна швидкість передачі інформації – 171,3 Кб/сек.

## 2.7.2. Основні служби Інтернету

Розглянемо основні сервіси, якими може скористатись будь-який користувач Інтернету за допомогою відповідного програмного забезпечення, яке взаємодіє із сервером, отримуючи від нього інформацію і відповідним чином її опрацьовуючи:

**електронна пошта (email)** - сервіс, який дозволяє пересилати текстові файли за довільною електронною адресою. Швидкість передавання текстових повідомлень значно вища від швидкості проходження звичайного листа. Кожне поштове повідомлення має заголовок, який містить розділи

To: (Куди)

From: (Звідки)

Subject: (Тема);

**звукова електронна пошта** – пересилання по мережі звукових повідомлень абоненту (якщо у вас та вашого абонента на ЕОМ встановлені “звукові карти” та мікрофони). При наявності цифрових відеокамер та швидких каналів зв'язку можливе передавання і зображень співбесідників (відеоконференції). Окремим видом електронної звукової пошти є IP-телефонія, за допомогою якої забезпечують значно нижчі тарифи на міжнародні

телефонні розмови, ніж у традиційних телефонних компаніях;

**IRC** (Internet Relay Chat) – обмін текстовими повідомленнями у режимі реального часу;

**телеконференції** - дискусійні групи з певної тематики. Тематика організована ієрархічно, наприклад **rec.music.folk** це телеконференція про народну музику. Головні розділи конференцій:

**comp** – все про комп'ютери;

**news** – новини;

**rec** – теми дозвілля, мистецтва;

**sci** – науково-дослідна діяльність;

**soc** – соціальні проблеми;

**talk** – дискусії з спірних питань;

**alt** – альтернативні погляди на речі;

**biz** - бізнес, пропозиції про ділову співпрацю;

**k12** – освіта, виховання дітей;

**misc** - різне.

**WWW (World Wide Web)** – в основу цього сервісу покладено принцип гіпертексту – гнучкої системи посилань, розташованих у тексті. Web-технологія, заснована на використанні гіпертексту, значно спростила роботу в Інтернеті, оскільки посилання на інші документи у **WWW** відбуваються автоматично;

**віддалений доступ (telnet)** – сервіс, який дозволяє користувачу запускати на виконання програми на інших машинах;

**передавання файлів (ftp)** – за допомогою цього сервісу користувач може копіювати файли з віддаленої ЕОМ на свої магнітні диски та переглядати зміст каталогів на віддаленій ЕОМ. Передавання файлів може здійснюватись у двійковому (binary) та текстовому (ASCII) режимах.

### 2.7.3. Засоби ОС **Linux** для роботи в Інтернет

Для роботи у глобальній мережі Інтернет в ОС **Linux** створена велика кількість програм. У збірку **Borsch** внесені такі програми:

переглядачі Веб-сторінок – **Konqueror**, **Mozilla Web Browser**, **Firefox Web Browser**  
поштові програми – **Kmail**, **Mozilla**.

Вибір конкретної програми для роботи з електронною поштою чи для навігації в Інтернеті залежить від вас. На рис. 132 - 135 подані копії екранів програм **Kmail**, **Konqueror**, **Mozilla Firefox**

Рис. 132. Вікно поштової програми **Kmail** у режимі перегляду вхідних листів

1- меню програми, 2 - панель інструментів, 3 – закла задинки програми, 4 – список отриманих листів, 5 – вікно перегляду тексту листа, 6 – нове повідомлення, 7 – перевірити (заглянути у) поштову скриньку, 8 – відповіді на отриманий лист, 9 - відповіді всім, 10 – переслати, 11 – відкрити книгу з адресами, 12 – перегляд непрочитаних листів

Рис. 133. Створення нового листа у програмі **Kmail**

1- меню програми, 2 - панель інструментів, 3 – рядок задання електронної адреси, 4 – адреси для надсилання копій листа, 5 – тема листа, 6 – вікно редагування тексту листа, 7 – відслати лист, 8 – відповіді на отриманий лист, 9 - додати до листа файл

Рис. 134. Вікно програми **Konqueror** (розпорядник файлів) з Веб-сторінкою фірми Epson

1 – меню програми, 2 – лінійка інструментів, 3 – рядок задання Веб-адреси (**www.epson.ru**), 4 – вікно відображення Веб-сторінки

Рис. 135. Вікно програми **Mozilla Firefox** з Веб-сторінкою проекту **Debian**

## 2.8. Системи керування базами даних (СКБД)

Бази даних є основною формою зберігання різноманітної інформації. Якщо доступ до записів баз даних можна отримати через локальну мережу, то таку СКБД називають мережевою. Доволі часто інформацію у базах даних розташовують у кількох таблицях, які зв'язують між собою. Такі СКБД називають реляційними. Коли записи бази даних фізично розташовані на кількох серверах (ЕОМ, які спеціально призначені для зберігання інформації), то таку СКБД називають розподіленою. На думку авторів, вивчення СКБД як інструментів для проектування і заповнення баз даних у школі є недоречним, оскільки це вимагає вагомих фахових знань, які шкільною програмою не передбачені (залишмо це вищій школі). Разом з тим сформувані правильні уявлення в учня про СКБД можна на основі спеціалізованих баз даних широкого призначення.

### 2.8.1. Правознавча система “**Національні акти України**”

Визначним прикладом інформаційної системи є програма “**Нормативні акти України**” ([www.nau.kiev.ua](http://www.nau.kiev.ua)). Системою **НАУ** можна безоплатно користуватись у онлайновому режимі або встановити локальну версію (щоправда, для ОС **Microsoft Windows**). Система містить велику кількість текстів правового характеру (нормативних актів), організованих у дерево документів (згадайте дерево каталогів магнітного диска). Текст знайденого документа можна роздрукувати, перенести у власну базу документів, зкопіювати у текстовий файл. Окрім пошуку документів у дереві документів, система підтримує пошук за ключовим словом, пошук за контекстом (у назві та у тексті), пошук за картотекою, складний пошук (з використанням логічних операцій – ТА, АБО). Крім безоплатних версій існують комерційні версії з додатковими функціями. Для користування онлайновою версією потрібно зайти на Веб-сторінку [www.nau.kiev.ua](http://www.nau.kiev.ua)

Рис. 136. Веб-сторінка системи **НАУ**

На рис. 137 подано вікно **НАУ** з інформацією “**Про сайт**” а на рис. 138 – вікно пошуку документів за їх типом

Рис. 137. Безкоштовні ресурси системи **НАУ**

Рис. 138. Вікно пошуку документів за їх типом

Безкоштовні ресурси ситеми **НАУ** надаються у “**Гостьовому режимі** “. Нижче подано текст довідки системи щодо цього режиму:

“У Гостьовому режимі Вам доступні:

- повні тексти україномовних документів у наступних розділах:
  - [нові нормативні акти](#) (розділ оновлюється щоденно);
  - нормативні акти вищих органів влади України (Верховна Рада України, Президент, КМУ, КСУ, ВСУ);

- нормативні акти міністерств та відомств України;
- чинні акти СРСР, УРСР;
- міжнародні угоди України;
- [словник законодавчих термінів](#);
- [нормативно-довідкові таблиці](#).

- 1 кБ перекладів документів російською мовою.
- 1 кБ перекладів документів англійською мовою.

Види пошуків, якими Ви можете скористатися у системі НАУ-онлайн:

[Пошук за реквізитами](#) (видавець, дата, номер, слова з назви, тексту)

[Пошук за типами документів](#) (за юридичною силою, видавником, датою)

[Пошук за офіційними виданнями](#) (за публікаціями в офіційних та відомчих виданнях)

[Пошук за класифікатором](#)

[Пошук за тематикою](#) “

### 3. Завдання до практичних занять з шкільного курсу інформатики

#### Вступ

У практикумі подані завдання до практичних занять із шкільного курсу інформатики. Перед виконанням практичного завдання слід ознайомитись із теоретичним матеріалом до розглядуваної теми, який подано у підручнику до цього курсу (друкованому або електронному). Для допуску до виконання практичного завдання потрібно відповісти на запитання вчителя, приблизний текст запитань подано після кожного завдання (у цьому розділі подано завдання, які автор розділу Злобін Г.Г. використовував на практичних заняттях з учнями у львівській школах №6 і №25). Вчитель інформатики може використати цей текст як основу для створення власних завдань до практичних занять (з обов'язковим посиланням на цей посібник). Створений текст доцільно оформити **pdf**-файлом і записати його у каталог **Home** на усі учнівські ПЕОМ. Для створення **pdf**-файл можна використати текстовий процесор **OpenOffice.orgWriter** (**Файл – Експорт як pdf**).

#### Завдання I. Початок і закінчення роботи в ОС **Linux**

Отримайте у вчителя ім'я користувача і пароль. Увімкніть ПЕОМ і зареєструйтесь у системі. Користуючись командою “**Вийти**” завантажте по черзі задані вчителем графічні оболонки.

Контрольні запитання:

1. Як здійснюється реєстрація в ОС **Linux**?
2. Як здійснюється вибір графічної оболонки для роботи з ОС **Linux**?
3. Як правильно вимкнути ПЕОМ під час роботи з ОС **Linux**?

#### Завдання II. Дії з файлами і каталогами в ОС **Linux**

У каталозі **Home** створіть два каталоги із заданими викладачем іменами. Користуючись функцією пошуку файлів розшукайте на ПЕОМ файли з такими розширеннями:

**txt**

**png**

**wav**  
**jpg**  
**sxw**

Скопіюйте по два файли із заданими розширеннями у перший каталог. Покажіть викладачу результати своєї роботи (1 бал). Перенесіть ці файли у другий каталог і знову покажіть викладачу результати своєї роботи (1 бал). Перенесіть другий каталог із файлами у перший каталог і знову покажіть викладачу результати своєї роботи (1 бал). Перейменуйте усі файли за правилом file1.tst, file2.tst, file3.tst і т.д. Покажіть викладачу результати своєї роботи (1 бал). Видаліть створені вами каталоги із файлами і спорожніть “Кошик”. Покажіть викладачу результати своєї роботи (1 бал).

Контрольні запитання:

1. Що таке файл?
2. Що таке каталог (тека, folder)?
3. Як позначаються дискові запам'ятовуючі пристрої у в ОС **Linux**?
4. Які дії можна виконувати із файлами?
5. Які дії можна виконувати із каталогами?
6. Що таке права доступу до файлу, каталогу і як їх переглянути?

Завдання III. Методи запуску програм в ОС **Linux** та маніпулювання вікнами програм

Запишіть у робочому зошиті стовпчиком усі методи запуску програм, якими можна користуватись в ОС **Linux** і отримайте від вчителя назви програм, з якими вам доведеться працювати. Запустіть ці програми і розташуйте вікна програм черепицею, каскадом, з перекриттям. Покажіть результати своєї роботи вчителю. Виконайте ці дії у заданих вчителем графічних оболонках.

Контрольні запитання:

1. Як змінити розміри вікна програми?
2. Як тимчасово прикрити вікно програми ?
3. Як закрити вікно програми?
4. Які перемістити вікно програми по екрану дисплея?
5. Що таке “віртуальні стільниці” і як ними користуватись?

Завдання IV. Опрацювання текстової інформації на ПЕОМ

Частина I. Набір і форматування тексту.

Користуючись текстовим процесором **OpenOffice.orgWriter**, наберіть заданий вчителем текст. Відформатуйте цей текст за вказівками вчителя і збережіть його у своєму домашньому каталозі.

Контрольні запитання:

1. Як змінити розмір шрифту, яким набирається текст?
2. Як вибрати шрифт, яким набирається текст?
3. Як виконати форматування тексту?
4. Як зберегти набраний текст?
5. Як відкрити текстовий файл?

Частина II. Використання таблиць у текстових документах

Користуючись текстовим процесором **OpenOffice.orgWriter** створіть таблицю із списком учнів класу або його частиною і внесіть у таблицю оцінки кожного учня за вказаний вчителем період. Користуючись формулами процесора **OpenOffice.orgWriter** для кожного учня обчисліть суму набраних балів і середній бал за заданий період.

Контрольні запитання:

1. Як вставити таблицю у текст?
2. Як задають розміри таблиці?
3. Як вставити додатковий рядок або стовпець таблиці?
4. Як видалити рядок або стовпець таблиці?
5. Як підрахувати суму чисел у рядку або стовпці таблиці?

Частина III. Використання графіки у текстових документах

Користуючись текстовим процесором **OpenOffice.orgWriter** створіть бланк школи і наберіть заданий вчителем текст офіційного листа директора школи.

Контрольні запитання:

1. Як вставити графічне зображення у текст?
2. Як задати розміри графічного зображення?
3. Як задають взаємне розташування тексту і графіки?

Частина IV. Використання машинних словників для перекладу тексту

Користуючись текстовим процесором **OpenOffice.orgWriter** машинним словником **Stardict** виконайте переклад заданого вчителем тексту. Збережіть переклад у домашньому каталозі і покажіть його вчителю.

Контрольні запитання:

1. Як запустити на виконання машинний словник **Stardict**?
2. Як задати напрямок перекладу?
3. Як вкопіювати отриманий переклад у вікно текстового процесора **OpenOffice.orgWriter**?

Завдання V. Опрацювання числової інформації на ПЕОМ

Частина I. Запис арифметичних виразів мовою Паскаль

Користуючись оболонкою **Algo** уведіть програму обчислення заданих вчителем арифметичних виразів. Передбачте у програмі вивід отриманих значень на екран дисплея. Перегляньте текст створеної програми українською і англійською мовами. Запустіть складену програму на виконання. Виконайте складену програму у покроковому режимі.

Контрольні запитання:

1. Як запустити на виконання оболонку **Algo**?
2. Як можна використовувати шаблони для створення тексту програми?
3. Як переглянути текст створеної програми українською мовою?
4. Як переглянути текст створеної програми англійською мовою?
5. Як запустити створену програму на виконання?
6. Як обчислити значення функції, заданої складним виразом?

## 7. Як виконати покрокове виконання програми у оболонці **Algo**?

### Частина II. Побудова складених логічних виразів у мові Паскаль

Уміння складати логічні вирази надзвичайно важливе для сортування інформації у табличних процесорах, системах керування базами даних, інформаційних і пошукових системах. У цьому завданні ви повинні записати складений логічний вираз для заданої вчителем області і перевірити його за допомогою програми **lv.pas**. На рис. 139 подано вікно оболонки **Algo** із програмою перевірки логічного виразу і результатами її роботи.

Рис. 139. Програма перевірки логічного виразу з результатами її роботи.

Контрольні питання.

1. Як записуються у Паскалі прості логічні вирази?
2. Як записуються у Паскалі складені логічні вирази?
3. Запишіть позначення операцій відношення.
4. Запишіть позначення логічних операцій.

### Частина III. Дослідження функцій. Отримання таблиці значень функції (табуляція функції)

Дослідження поведінки функцій проводиться досить часто. Не завжди це дослідження вдається провести аналітично. У цьому випадку корисну інформацію можна отримати з перегляду таблиці значень функції у деякому діапазоні значень аргумента  $[a,b]$ . Задача отримання таблиці значень функції - табуляція може бути проведена за наступним алгоритмом (тут для запису алгоритму використано псевдокод [псевдокод – це форма запису алгоритмів, у якій використовують україномовне написання ключових слів мови програмування, опускають опис змінних та використовують речення розмовної мови]):

#### **Програма табуляція\_функції;**

Описати змінні

#### **ПОЧАТОК**

{ задати діапазон та крок зміни аргумента }

**a:=** значення ;

**b:=** значення;

**N:=** значення ;

**dx:=(b-a)/N;**

{ для заданого діапазону зміни аргумента отримати таблицю значень функції }

**x:=a;**

**ПОКИ x<=b ВИКОНУВАТИ**

**ПОЧАТОК**

**y:=f(x);**

**вивести x,y**

**x:=x+dx;**

**КІНЕЦЬ;**

**КІНЕЦЬ.**

Завдання до роботи:

1. Для заданого вчителем вигляду функції та проміжка табуляції (таблиця 1) вдома записати



у робочий зошит текст програми мовою Паскаль.

2. Набрати та відлагодити написану програму.
3. Визначити два найближчих значення аргумента функції, між якими відбувається зміна знаку функції.

Контрольні питання:

1. Структура програми мовою Паскаль.
2. Стандартні типи змінних Паскалю.
3. Команда присвоєння у мові Паскаль.
4. Команда виводу інформації у мові Паскаль.
5. Команда повторення (циклу) з невідомою кількістю повторень у мові Паскаль.

Варіант	Функція $f(x)$	Проміжок табуляції
1	$3\sin\sqrt{x} + 0.35x - 3.8$	[2; 3]
2	$x^2/4 + x - 1.2502$	[0;2]
3	$x - 1/(3 - \sin(3.6x))$	[0;0.8]
4	$0.1x^2 - 5x\ln(x) - 1$	[1;2]
5	$\cos(2/x) - 2\sin(1/x) + 1/x$	[1;2]
6	$3x - 4\ln x - 5$	[2;4]
7	$x - 2 + \sin(1/x)$	[0.9;2]
8	$e^x - e^{-x} - 2$	[0;1.5]
9	$x + x^{1/2} + x^{1/3} - 2.5$	[0.4;1]
10	$\operatorname{tg}x - (\operatorname{tg}^3x + 1)/3 + 0.2\operatorname{tg}^5x$	[0;0.8]
11	$e^x + \ln(x) - 10x$	[3;4]
12	$\cos x - \exp(-x^2/2) + x - 1$	[1;2]
13	$\sin(\ln x) - \cos(\ln x) + 2\ln x$	[1;3]
14	$\arccos x - (1 - 0.3x^3)^{1/2}$	[0;1]
15	$(1 - 0.4x^2)^{1/2} - \arcsin x$	[0;1]

#### Частина IV. Дослідження функцій. Побудова графіка функції

Перегляд графіка функції дозволяє швидко отримати уявлення про поведінку функції. Для заданих вчителем функції, проміжку значень аргумента **[a;b]** та максимального і мінімального значень функції побудуйте графік функції у заданій області зміни аргумента за наступним алгоритмом.

**Програма графік\_f(x);**

**Тип tab=масив[1..200] із дійсна;**

**Змінна у:tab;**

**a,b,dx,x,макс\_u,мін\_u:дійсна;**

**i,tx,ty:ціла;**

**Початок**

{ задати проміжок [a,b], зміни аргумента x та крок зміни dx }

{ отримати таблицю значень функції f(x) }

{ задати максимальне макс\_u та мінімальне мін\_u значення функції }

{вибрати масштаб побудови графіка }

$ky:=300/(\text{макс\_}y-\text{мін\_}y);$

Для  $i:=1$  до  $N$  виконати

Початок

$tx:=i;$

$ty:=300 - ky*(y[i]- \text{мін\_}y);$

Крапка( $tx,ty$ );

кінець;

кінець.

Табуляція функції вже розглянуті нами, тому розглянемо останню частину алгоритму. Графік функції будується у частині вікна виконання розміром 200 точок по горизонталі та 300 точок по вертикалі. Коефіцієнт масштабу  $ky$  та формула для  $y$ -координати точки графіка  $ty$  вибрані таким чином, щоб мінімальне значення функції відображалось у нижній частині відведеного вікна ( $ty=300$ ), а максимальне – у верхній частині ( $ty=0$ ). Оскільки значення аргумента  $x$  беруться з рівномірним кроком  $dx$ , то в якості  $x$ -координати точки графіка у вікні виконання використовується її номер ( $tx:=i$ ), а  $y$ -координата визначається оператором

$ty:=300 - ky*(y[i]- \text{мін\_}y);$

Спочатку на екрані ставиться точка для першого значення функції ( $i=1$ ), далі другого ( $i=2$ ), третього ( $i=3$ ) і т.д. аж до останнього значення ( $i=N$ ).

Контрольні питання:

1. Як поставити крапку на екрані дисплея?
2. Як задати колір крапки?
3. Як скористатись довідковою системою **Algo** для отримання опису графічних процедур?
4. Які графічні процедури оболонки **Algo** ви знаєте??

Частина V. Розв'язання нелінійного рівняння поділом відрізка навпіл

Для заданого вчителем рівняння знайдіть його розв'язок поділом відрізка навпіл із заданою точністю.

Контрольні питання:

1. Як перевірити отриманий розв'язок рівняння?
2. Як здійснюється визначення розв'язку рівняння у методі поділу відрізка навпіл?
3. Чи можна знайти точний розв'язок рівняння поділом відрізка навпіл?

Частина IV. Обчислення суми елементів лінійної таблиці

У середовищі **Algo** реалізуйте алгоритм обчислення суми елементів лінійної таблиці для ряду чисел, заданих вчителем. Доповніть складену програму обчисленням середнього значення ряду чисел.

Контрольні запитання:

1. Методи опису табличної інформації.
2. Стандартні типи змінних Паскалю.

3. Команда присвоєння у мові Паскаль.
4. Команда виводу інформації у мові Паскаль.
5. Команда повторення (циклу) з відомою кількістю повторень у мові Паскаль.

#### Частина V. Обчислення мінімального значення (робота із лінійними таблицями)

У середовищі **Algo** реалізуйте алгоритм обчислення мінімального значення для ряду чисел, заданих вчителем.

Контрольні запитання:

1. Умовна команда Паскалю.
2. Опис цілих змінних у Паскалі.
3. Методи задання значення елементів таблиць (масивів) у мові Паскаль.
4. Команда виводу інформації у мові Паскаль.
5. Команда повторення (циклу) з відомою кількістю повторень у мові Паскаль.

#### Частина VI. Обчислення максимального значення (робота із лінійними таблицями)

У середовищі **Algo** реалізуйте алгоритм обчислення максимального значення для ряду чисел, заданих викладачем.

Контрольні запитання:

1. Умовна команда Паскалю.
2. Опис дійсних змінних у Паскалі.
3. Методи задання значення елементів таблиць (масивів) у мові Паскаль.
4. Команда виводу інформації у мові Паскаль.
5. Команда повторення (циклу) з відомою кількістю повторень у мові Паскаль.

#### Частина VII. Використання табличного процесора **OpenOffice.orgCalc** для опрацювання числової інформації

Користуючись табличним процесором **OpenOffice.orgCalc** виконайте завдання частин IV – VI. Упорядкуйте заданий ряд чисел по зростанню, спаданню. Для цього ряду чисел побудуйте стовпчикову діаграму.

Контрольні питання.

1. Як здійснюється уведення числових значень у табличному процесорі **OpenOffice.orgCalc**?
2. Як звернутись до функцій табличного процесора?
3. Як побудувати діаграму у середовищі табличного процесора?

#### Частина VIII. Заповнення таблицю обліку робочого часу працівників малого підприємства із погодинною оплатою праці у середовищі табличного процесора **OpenOffice.orgCalc**

У середовищі табличного процесора заповніть таблиць обліку робочого часу для 5-6 працівників малого підприємства і підрахуйте для кожного працівника суму відпрацьованих годин. Задайте ім'я "tabel" цього аркуша електронної таблиці і збережіть результати роботи у власному каталозі.

Контрольні питання.

1. Яка інформація може вноситися у комірки електронної таблиці?

2. Як підрахувати суму чисел рядка або стовпця електронної таблиці?
3. Як змінити назву аркуша електронної таблиці?

Частина IX. Формування відомості нарахування заробітної плати працівникам малого підприємства із погодинною оплатою праці у середовищі табличного процесора **OpenOffice.orgCalc**

На другому аркуші електронної книги сформууйте відомість нарахування заробітної плати із наступною структурою

<i>П.І.Б</i>	<i>Нараховано</i>	<i>Приб. податок</i>	<i>Внески у фонди</i>			<i>До видачі</i>
			<i>Пенсійний</i>	<i>Соц. страх.</i>	<i>Зайнятості</i>	
Бучко Ю.С.						
Гаврилюк І.М.						
Заяць М.І.						
Коненко П.С.						
Янків О.П.						
Всього						

Для числових значень цього аркуша використайте грошовий формат. Задайте ім'я "vidom" цього аркуша електронної таблиці і збережіть результати роботи у домашньому каталозі.

Контрольні питання.

1. Як підрахувати внесок праці вника у пенсійний фонд?.
2. Як у другому аркуші електронної таблиці можна використати числову інформацію із першого аркуша електронної таблиці?
3. Як можна вставити в електронну таблицю додаткові аркуші?

Частина X. Побудова колової діаграми розподілу коштів фонду оплати праці малого підприємства у середовищі табличного процесора **OpenOffice.orgCalc**

На третьому аркуші електронної таблиці сформууйте зведену таблицю виплат із фонду оплати праці малого підприємства із наступною структурою

До видачі	
Всього приб. податку	
Відрахування у пенсійний фонд	
Відрахування у фонд соц. страхування	
Відрахування у фонд зайнятості	

Врахуйте те, що відрахування в усі фонди складаються із двох частин: внеску працівників;

відрахувань від суми нарахованої заробітної плати (Всього нараховано) за нормативами – пенсійний фонд 32%, фонд соціального страхування - 4%, фонд зайнятості -1%.

На основі другого стовпця таблиці побудуйте колову діаграму розподілу коштів фонду оплати праці.

Контрольні питання.

1. Як побудувати діаграму у табличному процесорі?
2. Перерахуйте види діаграм, які можна будувати за допомогою табличного процесора.
3. Як можна видалити із електронної таблиці аркуш?
4. Як можна задати ім'я аркуша?
5. Як адресуються комірки на аркуші електронної таблиці?
6. Як адресуються комірки у електронній книзі, яка складається з багатьох аркушів?

## Завдання VI. Опрацювання графічної інформації на ПЕОМ

### Частина I. Створення простих зображень у графічному редакторі **KolourPaint**

Користуючись графічним редактором **KolourPaint** створіть план вашого кабінету інформатики. Збережіть створений план у домашньому каталозі.

Контрольні питання:

1. Що таке растрова графіка?
2. Поясніть призначення інструментів редактора **KolourPaint**.
3. Як виконати написи на рисунку?

### Частина II. Створення простих зображень у векторному графічному редакторі **OpenOffice.orgDraw**

Користуючись графічним редактором **OpenOffice.orgDraw** створіть план вашого кабінету інформатики. Збережіть створений план у домашньому каталозі. Порівняйте розміри файлів, створених растровим і векторним графічними редакторами.

### Частина III. Створення презентації у системі **OpenOffice.orgImpress**

Користуючись системою **OpenOffice.orgImpress** складіть презентацію на одну із тем:

1. Дії з файлами і каталогами в ОС **Linux**;
2. Методи запуску програм в ОС **Linux**;
3. Офісний пакет **OpenOffice.org** і його складові.

Для отримання графічних файлів для презентації використайте функцію захоплення графічного екрану редактора **Gimp** (**Файл – Захопити**).

## Завдання VII. Пошук інформації у глобальній мережі Інтернет і робота з електронною поштою

Частина I. Користуючись переглядачем Веб-сторінок увійдіть на Веб-сторінку системи **НАУ** і вкопіюйте два абзаци заданого вчителем документа. Збережіть вкопіюваний текст у текстовому файлі.

Частина II. Користуючись поштовою програмою скеруйте електронний лист з вкопіюваним текстом на задану вчителем електронну адресу.

Список рекомендованої літератури.

1. С. Апунович, Г. Злобін, О. Кустовінов, Т. Костюк Методичні вказівки щодо використання ОС Linux в школі. У двох частинах – Львів: Техноекс, 2002. Частина перша – 64 с., Частина друга – 117 с.,
2. Г. Злобін Персональний комп'ютер. Навчальний курс. II видання – Львів: Простір-М, 2006. - 294 с., рис.
3. Петрів В.Ф. Програмне забезпечення та методика вивчення програмування. - Львів: 2003. -152 с.