МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«УЖГОРОДСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ»
ІНЖЕНЕРНО-ТЕХНІЧНИЙ ФАКУЛЬТЕТ
КАФЕДРА КОМП'ЮТЕРНИХ СИСТЕМ ТА МЕРЕЖ

МЕТОДИЧНІ ВКАЗІВКИ І ЗАВДАННЯ
ДО ЛАБОРАТОРНИХ РОБІТ З КУРСУ

ОПЕРАЦІЙНІ СИСТЕМИ

для студентів 3-го курсу інженерно-технічного факультету
напряму підготовки «Комп'ютерна інженерія»

Ужгород – 2015

Методичні вказівки і завдання до лабораторних робіт
з курсу «Операційні системи»
для студентів 3-го курсу інженерно-технічного факультету,
напрям підготовки «Комп'ютерна інженерія».

Укладачі: Гапак О. М. – канд. пед. наук, доцент кафедри комп'ютерних систем та мереж.

Рецензент: Глебена М. І. – канд. фіз.-мат. наук, доцент кафедри системного аналізу та теорії оптимізації УжНУ.

Відповідальний за випуск – Король І.Ю., канд. фіз.-мат. наук, доцент, завідувач кафедри комп'ютерних систем та мереж.

Дані методичні вказівки розглянуто та схвалено на засіданні кафедри комп'ютерних систем та мереж, протокол № 1 від 31 серпня 2015 року та методичної комісії інженерно-технічного факультету, протокол № 1 від 15 жовтня 2015 року.

ВСТУП

Мета курсу – надання студентам знань та навичок щодо сучасних операційних систем, їх раціонального використання, а також практичних навичок ефективного використання сучасних операційних систем у процесі функціонування організації.

Завдання дисципліни – дати студентам базові знання з ідеології, побудови, сучасних операційних систем (ОС), їх функціональних характеристик, а саме структури та складу основних компонент ОС, їх функцій, взаємозв'язок їх характеристик з характеристиками технічних засобів та потреб, що виникають з боку різних категорій користувачів; формування навичок та уміння в питаннях інсталяції, настроювання та адміністрування операційних систем сімейства Linux та Windows.

У результаті вивчення дисципліни студент повинен знати і вміти: класифікувати операційні системи, маніпулювати основними командами та елементами файлової структури ОС Linux та Windows, управляти процесами мультипрограмної ОС, користуватися програмними оболонками ОС, оперувати механізмом вводу/виводу, користуватися мережевими можливостями ОС.

У даних методичних вказівках запропоновано ряд лабораторних робіт для вивчення ОС Linux.

ОС Linux – це багатокористувацька, багатозадачна, багатотермінальна операційна система із сімейства UNIX, під керуванням якої можуть одночасно виконуватися кілька задач. Вона призначена для роботи на серверах і робочих станціях, забезпечує підключення додаткових терміналів і допускає в цьому режимі використання графічних оболонок.

UNIX-сервери призначені для збереження й обробки великих обсягів інформації. Особливо ефективне використання UNIX-серверів при розподіленій обробці даних. Для цього розроблені системи розподілених обчислень у відповідності зі стандартом CORBA. До таких систем відносяться системи керування базами даних (СУБД типу Oracle, Informix), файли-сервери, FTP-сервери, WWW-сервери й ін., що підтримуються ОС Linux. У розподілених системах інформація може знаходитися на різних робочих станціях, різних дисках, програмні модулі можуть функціонувати на різних комп'ютерах, але система працює таким чином, що це складає єдине ціле. При обробці великих обсягів інформації використовується технологія клієнт – сервер, при якій користувач працює тільки з тією інформацією, що йому необхідна.

ОС Linux не зв'язана з конкретною моделлю комп'ютерів. Її ядро реалізоване мовою високого рівня (мова СІ), що дозволяє досить легко переносити цю систему з однієї платформи на іншу. Linux є безкоштовною ОС, яка вільно розповсюджується, зокрема через Internet, на основі відкритої ліцензії GNU GPL (GNU's Not Unix General Pubic License) у рамках Фонду безкоштовного програмного забезпечення FSF (Free Software Foundation). Багато прикладних програм для Linux можна отримати через Internet разом із їх вихідними текстами, це ж стосується і вихідного тексту самої Linux.

ЛАБОРАТОРНА РОБОТА № 1

Тема: Знайомство з ОС LINUX. Вивчення графічної оболонки KDE

Мета: Вивчити основи роботи з основними функціональними частинами графічної оболонки KDE, отримати навички налаштування KDE та створення найпростіших текстових та графічних документів KWord, Paint.

1. Струкутура ОС LINUX

Система включає наступні основні компоненти:

Ядро. Виконує функції керування пам'яттю, процесорами. Здійснює диспетчеризацію виконання всіх програм і обслуговування зовнішніх пристроїв. Усі дії, зв'язані з введенням/виведенням і виконанням системних операцій, виконуються за допомогою системних викликів. Системні виклики реалізують програмний інтерфейс між програмами і ядром.

Диспетчер процесів Init. Ініціалізує процеси, необхідні для нормальної роботи системи. Організує сеанси роботи користувачів, у тому числі, для віддалених терміналів, забезпечує завершення роботи системи.

Інтерпретатор команд Shell. Аналізує команди, що вводяться з термінала або з командного файлу, і передає їх для виконання в ядро системи. Shell є також мовою програмування, на якій можна створювати командні файли (shell-файли). При вході в ОС користувач одержує копію інтерпретатора shell як батьківський процес. Далі, після введення команди користувачем створюється породжений процес, що називають процесом-нащадком. Тобто після запуску ОС кожен новий процес функціонує тільки як процес - нащадок вже існуючого процесу. В ОС Linux є можливість динамічного породження і керування процесами.

Shell - інтерпретатор відповідно до вимог стандарту POSIX підтримує графічний екранний інтерфейс, реалізований засобами мови програмування Tcl/Tk.

Обов'язковим у системі є інтерпретатор Bash, цілком відповідний стандартові POSIX. У якості Shell може бути використана оболонка **mc** з інтерфейсом, подібним Norton Commander.

Мережний графічний інтерфейс X-сервер (X-Windows). Забезпечує підтримку графічних оболонок.

Графічні оболонки KDE, Gnome. Основними властивостями KDE є: мінімальні вимоги до апаратури, висока надійність, інтернаціоналізація. Gnome має більш розвинуті графічні можливості, але більш вимогливий до апаратних засобів.

Мережна підтримка NFS, SMB, TCP/IP. NFS - програмний комплекс PC-NFS (Network File System) для виконання мережних функцій. PC-NFS орієнтований для конкретної ОС персонального комп'ютера (PC) і включає драйвери для роботи в мережі і додаткові утиліти. SMB - мережна файлова

система, сумісна з Windows NT. TCP/IP - протокол контролю передачі даних (Transfer Control Protocol/Internet Protocol). Мережа по протоколах TCP/IP є невід'ємною частиною ОС сімейства UNIX. Підтримуються будь-які мережі, від локальних до Internet, з використанням убудованих мережних засобів.

Інструментальні засоби програмування. Основою засобів програмування є компілятор GCC або його експериментальні версії EGCS і PGCC для мов С и С++; модулі підтримки інших мов програмування (Obective C, Фортран, Паскаль, Modula-3, Ада, Java і ін.); інтегровані середовища і засоби візуального проектування: Kdevelop, Xwpe; засоби адаптації прив'язки програм AUTOCONFIG, AUTOMAKE.

Реєстрація користувача в системі

Для входу користувача з термінала в багатокористувацьку операційну систему LINUX необхідно зареєструватися. Для цього потрібно після повідомлення

Login:

увести системне ім'я користувача, наприклад, "student". Якщо ім'я задане вірно, виводиться запит на введення пароля:

Password:

Наберіть пароль "student" і натисніть клавішу *Enter*.

Якщо ім'я або пароль зазначені невірно, повідомлення *login* повторюється. Значення пароля перевіряється в системному файлі *password*, де знаходяться й інші дані про користувачів. Після введення правильної комбінації ім’я користувача – пароль з'являється вітання LINUX і запрошення:

student@linux:>

Ви одержали доступ до ресурсів ОС LINUX.

Вихід із системи:

exit - закінчення сеансу користувача.

reboot – перевантаження системи**.**

2. Налаштовування KDE

Центр керування (Control Center*)* – кнопка з зображенням гайкового ключа, складає основу всієї системи налаштувань KDE. В неї входить безліч панелей для всіляких компонентів робочого середовища і навіть деяких додатків KDE.

Більшість діалогових вікон мають кнопку виклику довідки. У найпростішому випадку – це контекстна довідка. Якщо виникли проблеми з пошуком необхідного діалогового вікна, на цій же панелі потрібно вибрати опцію Search (Пошук/Поиск).

Control Center

KDE надає широкі можливості по модифікуванню зовнішнього вигляду вікон і робочої області, включаючи відображення фону, ярликів, шрифтів тощо. Наприклад, можна керувати реакцією елемента на клік мишею, процесом завантаження і відображення обраних вікон, вибирати заставку (хранитель) екрана. Усі ці та багато інших можливостей може надати розглянута група

Control Center.

Для налаштування параметрів робочого столу і вікон варто вибрати опцію потрібного діалогу настроювання за назвою Desktop на дереві центру керування.

Зміна фону (тла)

Для зміни кольору фону або фонового візерунка робочого столу потрібно на дереві опцій центру керування послідовно вибрати Control Center – Appearance&Themes – Background. У результаті з'явиться діалогове вікно, що має три основні області:

• список віртуальних робочих столів;

• вікно попереднього перегляду;

• вікно настроювання параметрів.

Кожен віртуальний стіл у KDE має власні налаштування фону. Для кожного такого столу можна вибрати тло з одноколірною або двоколірною палітрою, а також фоновий візерунок. Якщо використовується фоновий візерунок, можна задати спосіб його відображення. Можна також вибрати кілька візерунків і автоматично переключатися між ними. Доступні і більш удосконалені опції, що дозволяють сполучити кольори і візерунки, а також підтримувати динамічні налаштування фону.

У процесі внесення змін в установки вони відображаються у вікні попереднього перегляду.

Віртуальні робочі столи

Робити налаштування параметрів віртуальних робочих столів у KDE можна в діалоговому вікні Control Center – Desktop – Multiple Desktops.

Покажчик Number of Desktops (Кількість робочих столів) показує, скільки віртуальних робочих столів доступно. Їхнє число може змінюватися в діапазоні від одного до шістнадцяти. Тут же можна задати назву для робочого столу, що потім буде відображено в списку вікон (Window List) або використано в настроюваннях панелі.

Заставка

Діалогове вікно вибору заставки екрану Appearance&Themes – Screensaver дозволяє вибрати анімацію і здійснити налаштування її параметрів. Опції налаштування бувають глобальні (опція установки часу запуску заставки) і індивідуальні — для кожного окремого зберігача. Діалогове вікно вибору має три основні секції:

• вікно попереднього перегляду;

• список програм – заставок;

• опції налаштування.

Необхідно вибрати назву потрібної програми з запропонованого списку.

Для налаштування параметрів необхідно клацнути на кнопці Setup (Налаштування) і в діалоговому вікні, що з'явилося, зробити установку потрібних характеристик.

Для установки інтервалу часу, через який буде запускатися заставка, потрібно ввести в поле опції Settings (Установки) величину даного інтервалу в хвилинах.

Налаштовування диспетчера вікон

За допомогою опцій Control Center Appearance&Themes – Desktop/Window behavior (Поводження/Поведение Вікон) центра керування можна встановлювати поводження диспетчера вікон. Опції у верхній частині діалогового вікна дозволяють виконати налаштування параметрів, що задають режим переміщення вікна і зміни його розмірів, а також визначають функціональність команди Maximize (Розгорнути).

Window behavior/Moving – меню установок розміщення вікна на екрані Placement (Розташування) дозволяє визначити місце на екрані, де буде відображатися вікно. Підтримуються такі методи.

• **Smart (Розумний)** — мінімізується перекриття між вікнами.

• **Cascade (Каскад)** — перше вікно відображається в лівому верхньому куті. Наступне вікно відображається зміщеним трохи вправо і вниз, так що вікна практично цілком перекриваються. І так далі.

• **Random (Довільний)** — вікна розташовуються на екрані в довільному порядку.

Метод одержання фокуса (тобто метод виділення окремих вікон або елементів) є, мабуть, індивідуальним методом настроювань KDE. За допомогою цього методу визначається, яке з відкритих вікон активне і які варто виконати дії при активізації вікна. Більш детально це виглядає так.

• **Click to focus (Передача фокуса щигликом).** Вікно одержує фокус (тобто стає активним) при щиглику на ньому мишею. При цьому вікно автоматично виводиться на перший план стосовно інших вікон. Такий метод використовується за замовчуванням.

• **Focus follows mouse (Фокус за мишею).** Вікно одержує фокус при безпосереднім звертанні до нього (це можна зробити за допомогою покажчика миші, використовуючи комбінацію клавіш <Alt+Tab> тощо). При цьому вікно може підніматися поверх інших вікон, а може і не підніматися. Переміщення покажчика миші на робочу область за межі вікна не означає втрату останнім фокуса. При виборі опції Auto Raise (Спливати автоматично) вікно буде спливати на екрані при переміщенні в його область курсору протягом декількох мікросекунд. Число цих мікросекунд установлюється за допомогою важільця Delay (Затримка). Якщо обрано опцію Click Raise (Спливати при щиглику), вікно буде підніматися поверх інших вікон при щиглику в будь-якій частині вікна. У іншому випадку така реакція вікна буде спостерігатися тільки при щиглику на його заголовку. Це винятково корисний метод передачі фокуса, оскільки дозволяє набирати текст в одному вікні й одночасно читати вміст іншого вікна, розташованого частково поверх зазначеного.

• **Focus Under Mouse (Фокус під мишею).** Вікно одержує фокус при будь-якому переміщенні на нього покажчика миші. При цьому комбінація клавіш <Alt+Tab> може і не допомогти.

• **Focus Strictly Under Mouse (Фокус тільки під мишею).** Вікно одержує фокус, тільки якщо покажчик миші знаходиться усередині вікна. Якщо покажчик миші знаходиться в робочій області, де немає вікон, жодне з вікон не одержить фокус.

Використання вікон

Відкрите вікно складається з наступних елементів.

Window menu (Меню керування вікном) - У лівому верхньому куті кожного вікна знаходиться піктограма маніпулювання вікном. При щиглику на ній з'являється меню, що містить команди за допомогою яких можна маніпулювати даним вікном. Maximize (Максимізувати) збільшить вікно до максимально можливого розміру. Minimize (Мінімізувати) зробить ваше вікно невидимим. Move (Перемістити) дозволяє пересувати вікно за допомогою миші. Size (Змінити розмір) дозволить вам збільшити або зменшити вікно. Shade – згорне вікно до заголовка. To desktop...(На робочий стіл) дозволить перевести вікно на інший робочий стіл. Виберіть робочий стіл, на який ви хочете перемістити це вікно. Вікно при цьому зникне. Для того щоб побачити його знову, виберіть ім'я на Лінійці задач, або клацніть на відповідну кнопку робочого столу на панелі KDE. Close (Закрити) закриє дане вікно. Always on Top – залишає вікно поверх усіх відкритих вікон.

3. Порядок виконання роботи

1. Запустіть файловий менеджер, користуючись командою „Виконати” із системного меню. Вивчіть структуру вікна, з’ясуйте які розділи є на вашому комп’ютері, занотуйте в звіт.

2. Розгляньте основні можливості пошукової системи, використавши команду „Пошук”. Здійсніть пошук файлів за маскою: *.htm, A???. Виконайте пошук файлів, які були створені за останній тиждень. Результати пошуку (повний шлях) записати в звіт.

3. Ознайомтесь з довідковою системою. Отримайте і запишіть інформацію про комбінації клавіш для виконання деяких дій (закрити, відкрити вікно; вимкнути комп’ютер; перемикання клавіатури; перехід по робочих столах; копія робочого столу та інше).

4. Виконайте деякі на лаштування графічної оболонки:

4.1. Створіть 5 робочих столів з різними фонами. Поміняйте тло, спочатку на одноколірний, а потім на двоколірний, далі вставте фонове зображення .

4.2. На робочому столі №3 встановіть заставку за власним бажанням, і режим очікування – 1 хвилина.

4.3. Налаштуйте переміщення вікон разом із їх вмістом.

4.4. Вивчіть можливості налаштування мови. В якому форматі відображаються числа, дата і час? Відповіді занотуйте у звіт.

4.5. Вивчіть налаштування миші. Задайте звуковий щиглик, що підтверджує натискання клавіші.

4.6.Вивчіть на лаштування монітору та клавіатури.

5. Запустіть калькулятор на робочому столі №1. Вивчіть його можливості та обчисліть: а) 25*12= б) 2^10= в) $\sin^2(90^0)$. Результати запишіть в зошит.

6. Запустіть програму текстового редактора на робочому столі №1 та програму графічного редактора на робочому столі №2.

7. Відкрийте текстовий редактор офісного пакету і наберіть наступний текст, що містить тему та мету даного лабораторного заняття, використовуючи два різних стилі за вашим вибором. Збережіть цей файл у домашньому каталозі користувача, закрийте редактор.

8. Відкрийте ваш домашній каталог користувача файловим менеджером, створіть у ньому каталог, скопіюйте ваш текстовий файл у цей каталог.

9. Створіть будь-який малюнок за допомогою графічного редактора, щоб у ньому були всі основні фігури (еліпс, коло, лінія, прямокутник, круг), не менш чотирьох кольорів.

10. Збережіть файл із малюнком у домашньому каталозі, закрийте графічний редактор.

11. Скопіюйте файл із малюнком у створений вами каталог.

12. Знайдіть на комп'ютері музичний файл та відеофайл за розширенням. Перегляньте їх.

4. Контрольні запитання

1. Що є компонентом робочого столу KDE?
2. Назвіть функції панелі робочого столу.
3. Які функції надає центр керування KDE?
4. Як викликати системне меню?
5. Назвіть головні команди системного меню.
6. Назвіть програми з групи Додаткові додатки системного меню.
7. Які програми є у групі Graphics системного меню?
8. Які програми є у групі Office системного меню?
9. Які програми є у групі Sound&Video системного меню?
10. Для чого слугує Центр керування KDE?
11. Назвіть деякі пункти налаштування з Центру керування KDE.
12. Яким чином користувач може змінити пароль у Linux?
13. Для чого слугує команда Виконати?
14. Для чого призначений пункт Недавні документи у системному меню?
15. Як можна у файловій системі відшукати файл за відомою назвою?
16. Що таке пошук за маскою? Як він виконується?
17. Як відшукати файл, якщо відомий час його створення?
18. Для чого призначена довідкова система?
19. Як користуватись довідковою системою?
20. Як зміни мову інтерфейсу KDE?
21. Як викликати калькулятор?
22. Як викликати графічний редактор?
23. Як зберегти створений файл у власному каталозі?
24. Що означає символ „*" у масці назви файлу?
25. Що означає символ „?" у масці назви файлу?

ЛАБОРАТОРНА РОБОТА № 2

Тема: Основи роботи з пакетом OpenOffice.org

Мета та зміст роботи:

1. Створення і редагування текстового документу OpenOffice.org Writer.
2. Створення електронної таблиці OpenOffice.org Calc.
3. Розробка графічного примітиву в OpenOffice.org Draw.
4. Створення презентації у OpenOffice.org Impress з використанням готових файлів.

1. Теоретичні відомості

OpenOffice.org – це вільний, багатофункціональний офісний пакет.

OpenOffice.org (OOo) одночасно є і програмним продуктом і об'єднанням добровольців, які забезпечують і підтримають програмне забезпечення. Кожний може вільно розповсюджувати OOo, із розробкою на основі відкритого коду.

OpenOffice.org включає в себе такі компоненти:

Writer – текстовий процесор. Це інструмент з великими можливостями для створення листів, книг, звітів, брошур та інших документів.

Можна вставляти різні об'єкти із інших компонент в документ Writer. Writer може експортувати файли в HTML, PDF і деякі версії файлів Microsoft Word.

Додатково до звичайних властивостей текстового редактора (перевірка орфографії, розстановка переносів, автозаміна та інше), Writer забезпечує важливі можливості:

- шаблони і стилі;
- методи макетування сторінок (включно рамки, стовпці, таблиці);
- вбудована або зв'язана графіка, електронні таблиці та інші об'єкти;
- експорт в формат PDF, включаючи закладки;
- інтеграція з базами даних.

Calc – електронна таблиця, що дозволяє обробляти і візуалізувати табличні дані. В електронну таблицю можна вводити дані, зазвичай числові, а потім маніпулювати даними для отримання певних результатів. Calc має сучасні засоби аналізу, побудови діаграм і можливості прийняття рішень. Calc може експортувати електронні таблиці в HTML, PDF.

Impress – програма для роботи з презентаціями. Вона забезпечує всі загальні засоби представлення мультимедіа, такі як спеціальні ефекти, анімація, засоби малювання. Він об'єднаний з розширеними графічними можливостями компонентів OOoDraw , Math.

Draw – представляє собою інструмент для малювання, що використовує векторну графіку. Він містить ряд сервісів, що дозволяють швидко створити всі види малюнків.

Math – редактор формул. Ви можете його використовувати для створення складних рівнянь, які включають знаки або символи, що недоступні в стандартних шрифтових наборах.

Переваги OpenOffice.org перед іншими офісними пакетами:

– Відсутність ліцензійної плати. OOo вільний для будь-якого використання і розповсюджується безплатно.

– Відкриті вихідні тексти. Можна розповсюджувати, копіювати і змінювати програмне забезпечення.

– Міжплатформовість. OOo працює на декількох апаратних архітектурах і під різними операційними системами, такими як Microsoft Windows, Mac OSX, Linux, SunSolaris.

– Інтеграція. Компоненти OOo добре інтегровані один з одним.

2. Створення і редагування текстового документу OpenOffice.org

Writer. Щоб створити новий файл у OpenOffice.org Writer виберіть команду Файл – Створити – Текстовий документ. Після створення документу часто виникає потреба у його редагуванні.

В OpenOffice.org Writer є деякі відмінності у порівнянні з MSWord. Наприклад, для виправлення помилок треба встановити курсор перед або після невірної букви і ввести вірну. Помилкову літеру (символ) слід видалити, скористувавшись клавішами <Del> або <BackSpace> (←).

Розбивання тексту на абзаци відбувається за допомогою клавіші <Enter>. Підводячи курсор до першої букви (символу) абзацу або до останньої букви (символу) попереднього речення, натискайте клавішу <Enter>.

Для виділення фрагментів тексту різними накресленнями, потрібно виділити слово, встановивши на ньому курсор і два рази натиснувши лівої кнопкою миші, і вибрати на контекстній панелі кнопку із зображенням потрібного накреслення (жирний, курсив, підкреслення, верхній чи нижній регістр).

Для виділення тексту в OOWriter є кілька альтернативних способів. Текст можна виділити посимвольно клавішами управління курсором, утримуючи попередньо натиснуту клавішу <Shift>. Натиснувши <Ctrl><Shift>, клавішами управління курсором можна виділяти текст з точністю до слова. Комбінація клавіш <Shift><Page Up> виділяє текст на сторінку вверх, а <Shift><Page Down> - на сторінку вниз. Комбінація клавіш <Ctrl><A> виділяє текст всього документа. Також весь текст можна виділити через пункт меню “Правка” → Виділити все. Можна виділити увесь рядок тексту, натиснувши три рази на слові з рядку. При натиснутій клавіші <Shift> клік лівою кнопкою миші робить виділення від позиції текстового курсору до позиції курсору миші. Натиснувши і утримуючи клавішу <Ctrl>, лівою кнопкою миші можна виділяти фрагменти тексту в різних місцях документу.

3. Створення електронної таблиці OpenOffice.org Calc

Щоб створити новий файл у OpenOffice.org Calc виберіть команду Файл – Створити – Електронну таблицю. Для уведення формул у панелі формул треба: 1) натиснути комірку, у яку треба вставити формулу; 2) натисніть значок

Функція на панелі формул, у рядку введення з'явиться знак рівності, і тепер можна уводити формулу; 3) після введення потрібних значень слід натиснути клавішу ВВЕДЕННЯ або кнопку **Прийняти** для вставки результату в поточний осередок. Якщо потрібно очистити рядок уведення, натисніть клавішу ESC або кнопку **Скасування.**

Можна вводити значення і формули прямо в комірки, навіть якщо не видний курсор уведення. Усі формули повинні починатися зі знака рівності. При редагуванні формули з посиланнями посилання і зв’язані комірки будуть виділені тим самим кольором. Параметр **Кольорові посилання** можна відключити в діалоговому вікні Сервис - Параметры - OpenOffice.org Calc – Вид. Якщо потрібно переглянути обчислення окремих елементів формули, виділіть відповідні елементи і натисніть клавішу F9. Приміром, у формулі =SUM(A1:B12)*SUM(C1:D12) виділите частину SUM(C1:D12) і натисніть клавішу F9 для перегляду проміжного результату для цієї частини. Якщо при створенні формули виникає помилка, у поточній комірці з'являється повідомлення про помилку.

4. Розробка графічного примітива в OpenOffice.org Draw

Щоб створити новий файл у OpenOffice.org Draw виберіть команду Файл – Створити – Малюнок. Для створення графічного примітива у OpenOffice.org Draw натисніть на кнопці відповідної групи примітивів панелі інструментів; потім, вибравши потрібний примітив зі списку піктограм, що відкривається, відпустіть кнопку. У результаті включається режим створення примітива, у якому потрібно вказати за допомогою миші розташування ключових точок і відстаней примітива. У різних примітивів різне число параметрів; так, у простої лінії лише два параметри, а в кривої необмежена кількість.

Основні інструменти для малювання розташовуються на панелі інструментів малювання:

Лінії і стрілки

Для створення лінії необхідно вказати початкову і кінцеву точку лінії на робочій області малюнка: початкова точка лінії задається лівою кнопкою миші; потім, не відпускаючи кнопку, встановити курсор на кінцеву точку лінії і відпустити кнопку. Лінія створена.

Прямокутники

Для побудови прямокутника потрібно вказати положення двох протилежних вершин прямокутника перша вказується натисканням лівої кнопки миші; потім, не відпускаючи її, підводиться курсор до другої точки і фіксується фігура, відпустивши кнопку.

Кола, еліпси, дуги, сегменти і сектори

Для створення кола й еліпса досить вказати габаритні розміри примітива двома точками натисканням, перетягуванням і відпусканням лівої кнопки миші. У випадку дуги, сегмента або сектора потрібно вказати ще дві точки на контурі примітива простим натисканням і відпусканням лівої кнопки.

Текст

Текст створюється простим натисканням лівої кнопки миші в потрібному місці аркуша; з'явиться рамка набору тексту з текстовим курсором. При створенні тексту, вписаного в рамку, спочатку задайте рамку двома точками натисканням >розтягуванням >відпусканням правою кнопкою, потім наберіть текст. Розмір шрифту буде автоматично підібраний так, щоб текст займав всю область зазначеної рамки. Легенда це рамка зі стрілкою, що звичайно використовується для пояснення якоїсь частини малюнка. Вона створюється натисканням розтягуванням відпусканням правою кнопкою миші; потім усередину рамки легенди можна вставити текст за допомогою подвійного натискання на рамці. При введенні тексту рамка легенди автоматично змінює розмір.

Криві Без'є, рисовані криві, багатокутники

Ґрунтуючись на тригонометричних рівняннях, французький математик й інженер П'єр Без'є створив особливий спосіб простого й у той же час гнучкого опису складних контурів для металорізальних машин, що використовувалися в автомобілебудуванні; цей спосіб одержав назву кривих
Без'є і завдяки своїй простоті й гнучкості згодом став одним з найважливіших методів комп'ютерної графіки. Криві Без'є будуються за декількома точками й напрямними лініями. Точки, за якими будується крива, називаються опорними точками; кожна з них характеризується двома відрізками, розташованими на відповідною дотичній до кривої Без'є в опорній точці. Довжина кожної з них задає крутість наступного або попереднього сегмента кривої, а кут дотичної задає напрямок в обидва боки від опорної точки. При створенні кривої в OpenOffice.org Draw послідовно вказуються її опорні точки за допомогою лівої кнопки миші. Якщо після натискання кнопки для створення опорної точки не відпустити кнопку, то можна задати кут і довжину напрямних; якщо ж кнопку не утримувати, то довжина напрямних буде нульова, і така точка буде кутовою. Напрямна першої опорної точки повинна бути зазначена, інакше операція скасовується. Подвійне натискання лівою кнопкою миші завершує малювання кривої. Рисована лінія також є кривою Без'є, тільки кількість опорних точок, величини і кути напрямних визначаються програмою автоматично. Для створення мальованої лінії потрібно, натиснувши й утримуючи ліву кнопку миші, намалювати бажану криву від руки.

Створення *багатокутника* базується на визначенні усіх його вершин. Другу вершину потрібно вказувати відпусканням натиснутої лівої кнопки миші, інакше операцію буде скасовано; інші вершини вказують звичайним натисканням лівої кнопки миші, а останню вершину подвійним клацанням. Так само, як і при створенні кривої, можна користуватися клавішами Alt і Shift для замикання багатокутника та побудови кутів, кратних 45 градусам, відповідно.

З'єднувальні лінії

Цей об'єкт створюється аналогічно звичайній лінії. Особливістю з'єднувача є здатність прив'язуватися до об'єктів, тому при створенні з'єднувача замість точки початку або кінця лінії можна вказати який-небудь об'єкт і програма сама підбере найкращу точку підключення з'єднувача до нього.

Тривимірні об'єкти

Будь який тривимірний об'єкт створюється визначенням його максимального розміру в одному з двох вимірів. Тривимірний об'єкт створюється у фіксованих пропорціях, що змінюються вже після його створення.

Автофігури

Дозволяють швидко намалювати яв основні фігури так і більш складні – зірок, фігурні символи, стрілки, блок - схеми та ін.

Ефекти

Ці інструменти використовуються для зміни форми, орієнтації і заливання об'єктів.

5. Створення презентації у OpenOffice.org Impress з використанням готових файлів

Щоб створити новий файл у OpenOffice.org Impress виберіть команду Файл – Створити – Презентацію... OpenOffice.org Impress– має практично всі можливості продукту Microsoft PowerPoint, сумісний з його файлами. У плані організації роботи практично ідентичний. Формування інформації на полі слайду відбувається так же, можна використовувати майстер презентації. Можна використовувати шаблони та такі допоміжні інструменти, як спрямовуючі для розміщення об'єктів, автоматичне “приклеювання” об'єктів до спрямовуючих, масштабування та ін. OpenOffice.org Impress має всі інструменти малювання, що використовуються у OpenOffice.org Draw і дозволяє задавати властивості для кожного об'єкта окремо. Крім того цей додаток дозволяє створювати динамічні презентації, які можна модифікувати, навіть коли презентація працює. Проте OpenImpress, хоч і має достатню кількість шаблонів розмітки, але у ньому практично відсутні шаблони графічного оформлення, що не дає можливості для швидкої генерації слайд - презентацій. Проте більшість цих проблем може бути вирішена завдяки підключенню модулів, на лаштувань і шаблонів, яких досить багато в Інтернет і розповсюджуються безкоштовно.

6. Порядок виконання роботи

1. Завантажте текстовий редактор OpenOffice.org Writer.

2. Наберіть текст без використання клавіші Enter:

Шановні панове! Запрошуємо вас на ювілейну презентацію комп'ютерної фірми “Мікроінком”. Ми працюємо на українському ринку багато років. Інформаційні продукти нашої фірми знають і полюбляють багато користувачів країни. Наші комп'ютери працюють без рекламацій! Будемо раді вас бачити. Запам'ятайте адресу і час u1085 нашої презентації – Комп'ютерна вулиця, д. 5, 18-00. Довідки за телефоном 123-45-67.

3. Відкрийте створений вами документ і відредагуйте його таким чином:

Ш А Н О В Н І П А Н О В Е!

Запрошуємо вас на ювілейну **презентацію**
комп'ютерної фірми “Мікроінком”.

Ми працюємо на українському ринку багато років. Інформаційні продукти нашої фірми знають і полюбляють багато користувачів країни.

Наші комп'ютери працюють без рекламацій!

Будемо раді вас бачити.

Запам'ятайте адресу і час нашої презентації:
Комп'ютерна вулиця, д. 5, 18^{00}год.

Довідки за телефоном 123-45-67

4. Збережіть текст у файл з назвою Text.odt у домашньому каталозі.

5. Підготуйте таблицю розрахунку кількості студентів, що склали іспит на відмінно, добре, задовільно, незадовільно, і студентів, що не з'явилися (табл. 1).

а) створіть новий текстовий документ;

б) створіть таблицю з 9 стовпців і 14 рядків із заголовком “Зведення про успішність студентів”;

г) встановіть ширину стовпця A (для рядків 2-14) – 0,7 см;

д) встановіть ширину стовпця B (для рядків 2-14) – 2,7 см;

е) встановіть ширину стовпця C (для рядків 2-14) – 1,69 см;

д) встановіть ширину стовпця D-I (для рядків 2-14) – 1,84 см;

ж) введіть текст у комірках відповідно до зразка;

з) занесіть результати у відповідні рядки і стовпці для підрахування кількості оцінок;

і) Збережіть документ у файлі Table.odt.

Таблиця 1

1	**Зведення про успішність студентів**								
2	A	B	C	D	E	F	G	H	I
3		Навчальна дисципліна	Група	Усього складало	Відм.	Добре	Задов.	Незадов.	Не з'явилося
4		Інформатика							
5	1.		133		12	10	6	3	1
6	2.		134		7	9	6	3	2
7	3.		135		9	8	3	5	3
8	4.		136		8	8	8	3	2
9		Усього:							
		Вища математика							
10	1.		133		8	12	10	1	1
11	2.		134		12	9	6	3	2
12	3.		135		12	8	3	5	3
13	4.		136		7	8	8	3	2
14		Усього:							

6. Проведіть такі розрахунки за формулами у даній таблиці:

а) число студентів кожної навчальної групи, що складали іспит за певною дисципліною;

б) загальне число студентів, що склали екзамен з кожної дисципліни на *відмінно, добре і т.д.;*

в) кількість всіх студентів, що склали екзамен з певної дисципліни.

Для того, щоб ввести формулу у комірку слід перенести до неї курсор і натисніть кнопку “Сума” на контекстній панелі. Утримуючи ліву клавішу миші, виділіть ті комірки, значення яких треба занести у формулу. Натисніть кнопку “Застосувати”. Якщо потрібно вставити іншу формулу, то у меню Вид треба вибрати підменю Панель інструментів і у вікні, що з'явилося, відмітити пункт Панель формул. Потрібну формулу можна вибрати серед списку формул, якщо натиснути на кнопку Формула.

7. Завантажте табличний процесор OpenOffice.org Calc.

8. Сформуйте таблицю екзаменаційною відомості за нижченаведеною формою. Встановіть відповідну ширину для кожного стовпця таблиці (табл.2).

Таблиця 2

Екзаменаційна відомість

Група № _____ Дисципліна _____

№ п/п	Прізвище, ім'я, по-батькові	№ залікової книжки	Оцінка	Підпис екзаменатора

9. Після закінчення оформлення таблиці введіть у неї постійні дані: порядкові номери (11), ПІБ студентів, номера залікових книжок та оцінки.

10. Після заповнення таблиці вихідними даними, треба розрахувати кількість оцінок (відмінно, добре, задовільно, незадовільно), кількість тих, хто не з'явився і загальну кількість отриманих оцінок.

Для того, щоб у відповідній колонці відбувався підрахунок кількості п'ятірок, четвірок та ін., треба у неї ввести формулу (для прикладу =COUNTIF(E7:E12;5) – для п'ятірок, =COUNTIF (E7:E12;4) – для четвірок, =COUNTIF(E7:E12;3) – для трійок, =COUNTIF(E7:E12;2) – для двійок, =COUNTIF(E7:E12;“н/з”) - для не з'явилися, діапазон значень E7:E12 вводите відносно вашого прикладу).

11. Перейменуйте перший лист як “Екзамен1”. Скопіюйте перший лист і вставте дві його копії, назвавши "Екзамен2", "Екзамен3".

Для копіювання і вставки листа слід встановити курсор на імені поточного листа і викликати контекстне меню правою кнопкою миші. У контекстному меню вибрати параметр Перемістити/скопіювати, поставити галочку Копіювати і зі списку Вставити перед вибрати Лист 2. Щоб вставити ще лист треба повторити наведені дії.

12. На новому аркуші, який назвіть “Стипендія”, підготуйте для групи відомість призначення студентів на стипендію за результатами екзаменаційної сесії за нижченаведеною формою (табл.3).

Таблиця 3

ВІДОМІСТЬ ПРО ПРИЗНАЧЕННЯ СТУДЕНТІВ НА СТИПЕНДІЮ **Група № _____**

Мінімальний розмір стипендії

№ п/п	Прізвище, ім'я, по-батькові	Середній бал	Кіл-ть складених іспитів	Стипендія

13. Скопіюйте з першого аркуша список групи у другий.

Для введення формули обчислення середнього значення треба встановити курсор у потрібну комірку і викликати Майстер функції на Панелі формули. Виберіть у категорії Статистичні, функцію AVERAGEA подвійним натисканням лівої кнопки миші. Встановіть

курсор у полі Значення 1, натисніть на назві листа Екзамен 1 і виділіть комірку із оцінкою першого студента за перший екзамен. Встановіть курсор у полі Значення 2, натисніть на назві листа Екзамен 2 і виділіть комірку із оцінкою першого студента за другий екзамен. Встановіть курсор у полі Значення 3, натисніть на назві листа Екзамен 3 і виділіть комірку із оцінкою першого студента u1079 за третій екзамен. Натисніть кнопку **Продовжити.**

14. Введіть формулу розрахунку кількості екзаменів для першого студента, використовуючи функцію COUNT. Скопіюйте формулу у нижні комірки.

15. Введіть формулу для обчислення розміру стипендії першого студента. Ця формула повинна мати такий вид:

=IF(AND(D8>=4,5;E8=3);C5*1,5;IF(AND(D8>=3;E8=3);C5;0))

Увага!

1. У структурі формули є вкладені функції AND(), IF(). Для введення цих функцій треба скористатися Майстром функції, що знаходиться на Панелі формули.
2. При наборі формули автоматично розстановлюються круглі дужки і розділювачі – крапка з комою.
3. Знак $ позначає, що посилання на цю комірку при копіюванні не буде змінюватися.
4. У процесі набору формули постійно порівнюйте її з виразом, що наведений.
5. Якщо після вводу формули з'явиться синтаксична помилка, то слід перевірити кількість дужок, наявність розділювачів. Технологія введення формули:

- встановіть курсор у потрібній комірці
- викличте Майстер функцій, виберіть у категорії Логічні, подвійним натиском виберіть функцію IF, курсор буде знаходитися у середині круглих дужок; подвійним натиском виберіть функцію AND, курсор знаходиться у середині круглих дужок; натисніть у комірці, де показаний середній бал студента і з клавіатури введіть >=4,5
- у Майстрі функцій встановіть курсор у пункт Логічний вираз 2 і аналогічно сформуйте вираз, який вказує необхідну кількість складених екзаменів (число 3) - через крапку з комою запишіть другу частину виразу аналогічним чином - натисніть Продовжити

16. Скопіюйте формулу у нижні комірки.

17. Перевірте працездатність таблиці: змініть числа, розмір стипендії.

18. Збережіть документ під назвою Session.ods.

Такий вигляд повинні мати формули у трьох правих колонках таблиці обліку стипендії (рис. 1). Закрийте документ.

	Середній бал	***Кількість складених іспитів***	***Стипендія***
	=AVERAGEA('Екзамен 1'.E7;'Екзамен 1_2'.E7;'Екзамен 1_3'.E7)	=COUNT('Екзамен 1'.E7;'Екзамен 1_2'.E7;'Екзамен 1_3'.E7)	=IF(AND(D8>=4,5;E8=3); C5*1,5;IF(AND (D8>=3;E8=3);C5;0))
	=AVERAGEA('Екзамен 1'.E8;'Екзамен 1_2'.E8;'Екзамен 1_3'.E8)	=COUNT('Екзамен 1'.E8;'Екзамен 1_2'.E8;'Екзамен 1_3'.E8)	=IF(AND(D9>=4,5;E9=3); C5*1,5;IF(AND (D9>=3;E9=3);C5;0))
	=AVERAGEA('Екзамен 1'.E9;'Екзамен 1_2'.E9;'Екзамен 1_3'.E9)	=COUNT('Екзамен 1'.E9;'Екзамен 1_2'.E9;'Екзамен 1_3'.E9)	=IF(AND (D10>=4,5;E10=3); C5*1,5;IF(AND (D10>=3;E10=3);C5;0))

Рисунок 1 – Формули

19. Створіть діаграму, яка містить дані про середній бал студентів вашої групи. Обов’язково оформіть заголовок, легенду, підписи осі X та У, цифрові значення.

20. Завантажте OpenOffice.org Draw.

21. Створіть малюнок за зразком (рис.2).

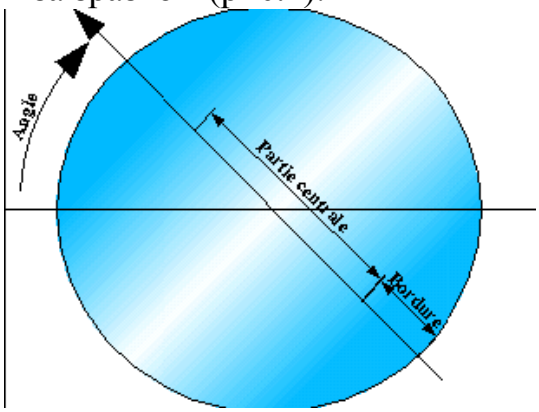

Рисунок 2 – Круг

22. Створіть тривимірне тіло обертання двовимірного креслення навколо вісі.

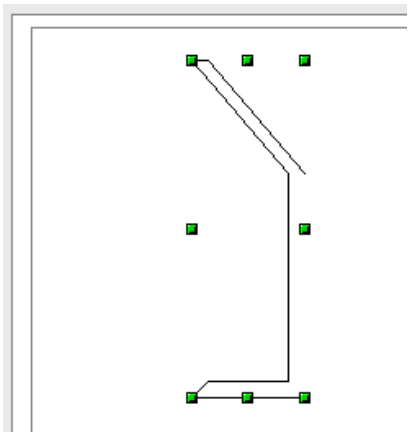

Рисунок 3– Базове креслення для створення тривимірного об'єкта

Створюєте за допомогою інструменту Багатокутник 45^0, що знаходиться у групі інструментів Об'єкт кривих, базовий профіль. Він може бути як замкнутим, так і відкритим.

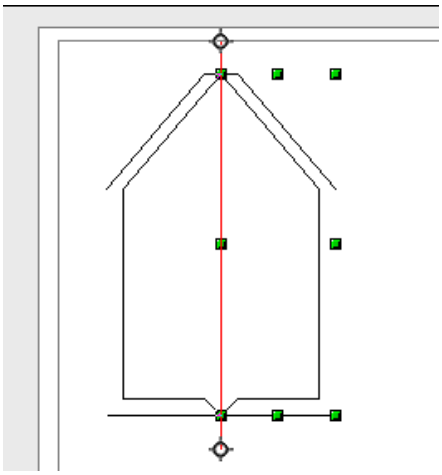

Рисунок 4 – Початкова вісь обертання

Натиснувши на кнопці "У тривимірне тіло обертання», що знаходиться у групі інструментів "Ефекти», ви додаєте вісь червоного кольору з ручками на кінцях. Кінцевий профіль вказаний з іншого боку вісі, щоб ви бачили, що вийде у результаті.

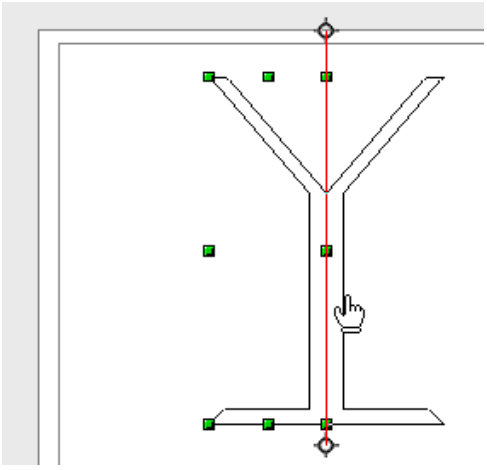

Рисунок 5–- Переміщення вісі симетрії

Щоб отримати креслення бокалу, треба перемістити вісь обертання, перемістивши її на інший бік профілю за ручки на кінцях. Кінцева форма вийде навколо вісі обертання

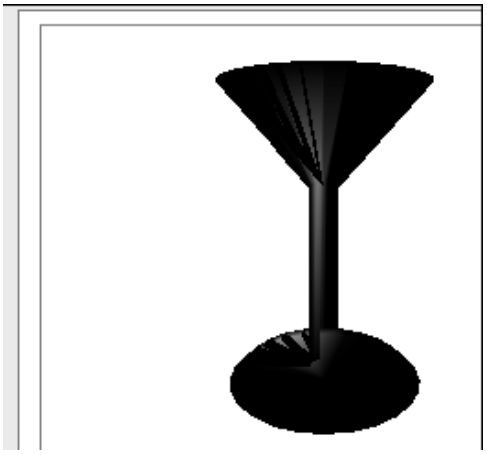

Рисунок 6 – Тінь тривимірного об'єкта

Щоб побачити результат, натисніть двічі на двовимірному об'єкті, який щойно був створений.

Рисунок 7 – Застосування прозорості ідо тривимірного об'єкту

Щоб зробити об'єкт більш реальним застосуйте синій колір і ефект прозорості, зробивши це за допомогою команди контекстного меню об'єкта Область.

23. Збережіть малюнок у файлі з назвою Mal.odg.

24. Завантажте OpenOffice.org Math. Наберіть задані формули:

$$x_{k+1} = x_k - \frac{f(x_k)(x_k - c)}{f(x_k) - f(c)}, \text{ де } c = \begin{cases} a, & f(a) \cdot f''(a) > 0 \\ b, & f(b) \cdot f''(b) > 0 \end{cases},$$

$$\cos\varphi = \frac{l_1 l_2 + m_1 m_2}{\sqrt{l_1^2 + m_1^2} \cdot \sqrt{l_2^2 + m_2^2}}.$$

$$\frac{(x - x_0)^2}{a^2} - \frac{(y - y_0)^2}{b^2} = 1.$$

25. Збережіть документ у файлі з назвою Formul.odg.

26. Завантажте OpenOffice.org Impress. Створіть презентацію заданої вам тематики з шаблону OpenOffice.org Impress:

– виберіть у меню Файл команду Створити – Презентацію,

– у вікні “Майстер презентації”, що відкрилося, виберіть Тип презентації – з шаблону, у списку, що випадає, виберіть “Фони презентацій” і у текстовому полі, що розташоване нижче виберіть той фон, що вам найбільше сподобався; натисніть кнопку “Далі” - у наступному вікні Майстра презентацій виберіть тип слайду <Оригінал> і спосіб відображення презентації – Оригінал; натисніть кнопку “Далі”

– наступний крок створення презентації з шаблону – вибір способу переходу від одного слайду до іншого, для цього слід вибрати такі параметри – Без ефектів, швидкість (Середня), За замовчуванням; натисніть кнопку “Далі” - у вікні, що з'явилося у полі “Ім'я або назва вашої організації” введіть – „Ужгородський національний університет, кафедра комп’ютерних систем та мереж, у полі “Тема” введіть «Лабораторна робота №2”, у полі «Сформулюйте ваші довгострокові плани» - «П.І.Б». Натисніть кнопку “Далі”. Відмітьте галочкою «Додати список слайдів» і натисніть кнопку вікні “Готово”.

27. На другому слайді розмістіть текст із файлу Text.odt, а на третьому таблицю з файлу Table.odt

28. На четвертому слайді розмістіть одну електронну таблицю з файлу Session.ods і діаграму.

29. На п’ятому – малюнки з Mal.odg.

30. На шостому – помістіть формули набрані в Math.

31. Виберіть параметри зміни (переходу) слайдів на екрані, завдавши звукові та анімаційні ефекти, а також час в автономному режимі.

32. Збережіть слайд-фільм у своїй робочій папці у двох форматах – презентації та демонстрації (Експорт (меню Файл) і зберегти документ у HTML-форматі або у форматі .swf). Закрийте програму.

Результати роботи.

За результатами роботи у домашньому каталозі має бути файл Table.odt, Text.odt, Session.ods, Mal.odg, Formul.odg і два файли презентації.

ЛАБОРАТОРНА РОБОТА № 3

Тема: ОС Linux. Текстовий режим функціонування

Мета: Ознайомлення з основними командами Linux, отримання навичок їх використання.

1. Текстовий режим роботи користувача в ОС Linux

Досі розглядалася робота користувача у графічній оболонці ОС Linux. Окрім графічного, є ще текстовий режим (інші терміни: режим терміналу, консолі) функціонування ОС Linux. Багато користувачів, а тим паче спеціалістів, надають йому перевагу. Є низка корисних програм та команд, які можуть функціонувати або лише у текстовому режимі, або як у текстовому, так і у графічному режимах. Це пов'язано з тим, що із самого початку Linux розроблялась як система, що успадковувала головні риси ОС Unix, яка працює здебільшого у текстовому режимі.

Під текстовим режимом розуміють введення команд користувачем лише з командного рядка, для чого їх потрібно спочатку набрати на клавіатурі. Прикладом консольної команди для запуску програми менеджера файлів MC є **mc**. Розглянемо три варіанти запуску сеансу роботи ОС Linux у текстовому режимі:

- реєстрація у текстовому режимі (Session — Failsave);
- використання спеціальних програм-емуляторів терміналу;
- переведення на іншу віртуальну консоль.

Щоб запустити текстовий режим з графічної оболонки, потрібно за допомогою пункту меню Виконати ввести назву відповідної програми, наприклад, xterm, konsole тощо. Програма konsole є стандартною для KDE.

В ОС Linux реалізована можливість одночасної роботи на одному комп'ютері декількох користувачів — це багатокористувацька система. Завдяки концепції віртуальних консолей користувачам не потрібно кожного разу перереєстровуватись. Для переходу на консоль з номером n слід натиснути Ctrl + Alt + Fn. Під час переходу на іншу консоль користувач автоматично потрапляє у текстовий режим, де необхідно увести свій логін (localhost login) та пароль (Password). Після закінчення сеансу роботи потрібно виконати логоф, увівши команду logout. Для повернення у графічну оболонку треба натиснути Alt + F8. У текстовому режимі можна виконувати практично усі дії, я у графічному. Однак для цього потрібно знати відповідні команди. Оскільки команд є багато і їх усіх запам'ятати складно, варто користуватися командою man <назва команди>, щоб [мати довідку про призначення, синтаксис та дію тієї чи іншої команди. Щоб запустити набрану на клавіатурі команду необхідно натиснути на клавішу Enter. Команда негайно буде виконана. Якщо в тексті команди допущено помилку, система виведе повідомлення: команда не знайдена (command not found). Зауважимо, що команди набирають у командному рядку, що містить запрошення системи у вигляді символу „$".

Формат команди наступний:

ім’я команди [аргументи][параметри][метасимволи].

Ім’я команди може містити будь-яке допустиме ім’я файлу; аргументи – одна або декілька букв із знаком мінус (-); параметри – значення, що передається для обробки; метасимволи інтерпретуються як спеціальні операції. В квадратних дужках вказуються необов’язкові частини команд. Назва команди складається з декількох малих латинських літер.

2. Основні команди ОС Linux

1. Введіть команду **echo**, яка видає на екран свої аргументи:

echo good morning

і натисніть клавішу Enter. Нам екрані з’явиться привітання „good morning” – аргумент команди **echo**. Командний інтерпретатор *shell* викликав команду **echo** у виді програми на мові СІ і передав їй аргументи. Після цього інтерпретатор вивів знак – запрошення. Синтаксис команди **echo:**

echo [-n][arg1][arg2]…

Команда поміщає в стандартний ввід свої аргументи розділені пропусками і завершуються символом переводу рядка. При наявності прапорця –n символ переходу рядка виключається.

2. **who [am i] –** отримання інформації про користувачів. Відповідь представлена у таблиці, що містить наступну інформацію: ідентифікатор користувача; ідентифікатор терміналу; дата підключення; час підключення.
3. **date –** визначення поточної дати і часу або їх зміна.
4. **write, mesg –** передача повідомлень іншому користувачу.
5. **mail –** відправка та читання поштових повідомлень.
6. **cal [[**місяць**]** рік**] –** відобразити календар дат.
7. **man** <назва команди> **–** виклик електронної довідки про вказану команду. Вихід із довідки при натисканні **Q**. Команда **man man** дає інформацію про те, як користуватись довідкою.
8. **clear –** очистка екрану.
9. **df** – виведення даних про розподіл дискового простору**.**
10. **free** – виведення повідомлення про розподіл пам’яті.
11. **mc** – запуск менеджера файлів.
12. **passwd** – змінити пароль користувача.
13. **history** – виводить на екран список раніше використаних команд.
14. **logout** – вихід із системи.
15. **reboot –** перевантаження системи.
16. **cat –** об’єднання і вивід вмісту файлів на екран.
    cat >file – створює новий файл із назвою file.
    cat file1>file2 – file1 копіюється у file2.
    cat file1 file2>result – вміст file2 буде добавлений в file1 і результатом буде result.
    cat file1>>file2 – вміст file1 буде добавлений в кінець file2.
17. **more –** вивід вмісту файлу.
18. **mkdir, rmdir –** створення і видалення каталогу.

За допомогою rmdir не можна видалити непустий каталог, але це можна зробити за допомогою команди rm –r із поточного каталогу буде вилучене все, включно і підкаталоги.

19. **ls –** вивід детальної інформації про файли та каталоги.

Кожний файл в ОС Linux має свій ідентифікаційний номер, наприклад, 1230, який називають *індексним дескриптором*, під яким файл реєструється у системі. Отримати на екрані індексні дескриптори можна командою **ls -i.**

ls **[-**a] [ім'я]

-a – вивести список всіх файлів, в тому числі і захованих.

-d – якщо аргумент є каталогом, то виводиться лише його ім'я, а не його вміст.

-c – вивід в декілька колонок з сортуванням по колонках.

-m – вивід у вільному форматі, імена файлів розділяються комами.

-l – вивід списку файлів в розширеному форматі.

-t – сортування по часу модифікації.

-s – виводить розміри кожного файлу.

20. **pwd –** вивід на екран повного імені поточного каталогу.

21. **cd –** зміна поточного каталогу.

22. **grep[-**v] [шаблон пошуку] [ім'я] **–** пошук файлів в системі каталогів

- v – виводяться рядки, що не містять шаблон пошуку.

-c – виводить тільки кількість рядків, що містять або не містять шаблон.

-i – при пошуку не розділяються малі та великі букви.

-l – виводяться тільки імена файлів, що містять вказаний шаблон.

23. **cp –** копіювання файлів

cp file1 file2 – копіювання файлів з перейменуванням.

24. **mv –** переміщення і перейменування файлів.

25. **rm** <ім'я файлу> **–** вилучення файлу.

26. **vi –** виклик текстового редактору.

27. **sort –** сортування вхідних файлів і вивід результату на екран.

28. **wc** <ім'я файлу> **–** визначення числових параметрів файлу (кількість рядків, слів, символів).

29. **chmod –** змінює режим доступу до файлів.

30. **zip, unzip –** архівувати (розархівувати) файли.

31. **file –** визначити тип файлу.

32. **lp –** вивести файл на принтер.

33. **ln –** організація посилань на файл**.**

В ОС Linux виділяють два види посилань: жорсткі та символічні. **Жорсткі** посилання можна створити лише в текстовому режимі за допомогою команди **ln <назва файлу> <назва посилання>**. Вони мають той самий індексний дескриптор, що й файл. Отримати список усіх жорстких посилань можна командою **ln -i**. Система веде облік кількості жорстких посилань на файл і відображає відповідне число в таблиці детальних властивостей файлу.

Посилання, які ви створювали у графічній оболонці, є символічними. **Символічне** посилання – це окремий короткий файл, який містить адресу того

файлу, на який воно вказує. У текстовому режимі символічне посилання можна створити за допомогою команди **ln -s <назва файлу> <назва посиланнях>.**

Відмінність між посиланнями така: якщо файл-оригінал перемістити на інше місце у файловій системі, то його символічні посилання не функціонуватимуть без переналаштування, а жорсткі функціонуватимуть далі. Якщо вилучити файл-оригінал із системи, то символічні посилання не функціонуватимуть зовсім, а жорсткі будуть. Лічильник обліку жорстких посилань зменшиться на одиницю. Власна назва файлу трактується як його (перше) жорстке посилання. Отже, файл-оригінал буде вилучено із системи після вилучення його останнього жорсткого посилання.

34. **kill –** переривання процесу.

35. **jobs –** список завдань, що виконуються.

36. **ps [al] [number]** - команда для виводу інформації про процеси:

-a - вивід інформації про всі активні процеси, запущених з вашого термінала;

-l - повна інформація про процеси;

number - номер процесу.

3. Групування команд

Групи команд або складних команд можуть формуватися за допомогою спеціальних символів (метасимволів):

& - процес виконується у фоновому режимі, не чекаючи закінчення попередніх процесів;

? - шаблон, поширюється тільки на один символ;

* - шаблон, поширюється на всі символи, що залишилися;

| - програмний канал - стандартний вивід одного процесу є стандартним введенням іншого;

> - переадресація виводу у файл;

< - переадресація введення з файлу;

; - якщо в списку команд команди відокремлюються одна від одної крапкою з комою, то вони виконуються одна за одною;

&& - ця конструкція між командами означає, що наступна команда виконується тільки при нормальному завершенні попередньої команди (код повернення 0);

|| - наступна команда виконується тільки, якщо не виконалася попередня команда (код повернення 1);

() - групування команд у дужки;

{ } - групування команд з об'єднаним виводом;

[] - вказівка діапазону або явне перерахування (без ком);

>> - додавання умісту файлу в кінець іншого файлу.

Приклади

who | wc - підрахунок кількості працюючих користувачів командою **wc** (word count - рахунок слів);

cat text.1 > text.2 - уміст файлу text.1 пересилається у файл text.2;

mail student < file.txt - електронна пошта передає файл file.txt усім користувачам, перерахованим у командному рядку;

cat text.1,text.2 - проглядаються файли text.1 і text.2;

cat text.1 >> text.2 - додавання файлу text.1 у кінець файлу text.2;

cc primer.c & - трансляція CI - програми у фоновому режимі. Ім'я виконуваної програми за замовчуванням a.out.

cc -o primer.o primer.c - трансляція CI-програми з утворенням файлу виконуваної програми з ім'ям primer.o;

rm text.* - видалення усіх файлів з ім'ям text;

{cat text.1; cat text.2} | lpr - перегляд файлів text.1 і text.2 і вивід їх на друк;

4. Порядок виконання роботи

1. Ознайомтесь з теоретичною частиною лабораторної роботи.
2. Увійдіть у текстовий режим. Зареєструйтесь у системі.
3. Перегляньте, хто на цей момент працює в мережі за допомогою команди **who**.
4. Отримайте інформацію про себе за допомогою команди **whoami.**
5. Надішліть повідомлення за допомогою команди **write** <ім’я>+Enter+текст повідомлення+**Ctrl+D.**
6. Перевірте поточну дату за допомогою команди **date.**
7. Виведіть календар за поточний місяць – команда **cal** <номер року>**.**
8. Дізнайтесь, якого дня тижня ви народилися – команда **cal** <номер місяця номер року>**.**
9. Отримайте інформацію про розподіл дискового простору – команда **df.**
10. Проаналізуйте розподіл пам’яті на вашому комп’ютері – команда **free**.
11. Запустіть менеджер файлів – команда **mc.** Познайомтесь з роботою в ньому.
12. Очистіть екран – команда **clear.**
13. Якщо можливо, змініть пароль користувача.
    Уведіть команду **passwd**. Введіть старий пароль та новий. Якщо система дозволить , то пароль буде змінено і ви отримаєте повідомлення.
14. Визначте шлях до поточного каталогу – команда **pwd.**
15. Перегляньте вміст каталогу – команда **dir**.
16. Створіть у каталозі **home** каталог **mykat –** команда **mkdir mykat.**
17. Активізуйте його – команда **cd mykat.**
18. У **mykat** створіть каталог **mykat1** та активізуйте його.
19. У даному каталозі створіть текстовий файл text.txt – **vi text.txt.** Натисніть клавішу **„i”** введіть назву закладу в якому навчаєтесь. Для збереження тексту та виходу з текстового редактору натисніть „**Esc**” для переходу в командний режим, потім двокрапку „ **:** ” для переходу в режим останнього рядка, далі **w**(записати) **q**(вийти).

20. Перегляньте вміст файлу – **less text.txt** або **more text.txt.** Проаналізуйте спосіб виведення тексту.
21. Вилучіть даний файл – **rm text.txt.**
22. Спробуйте вилучити каталог **mykat** за допомогою команди **rmdir.** Як правильно вилучити даний каталог?
23. Створіть дану структуру файлової системи, де N – номер індивідуального варіанту (рис.1).

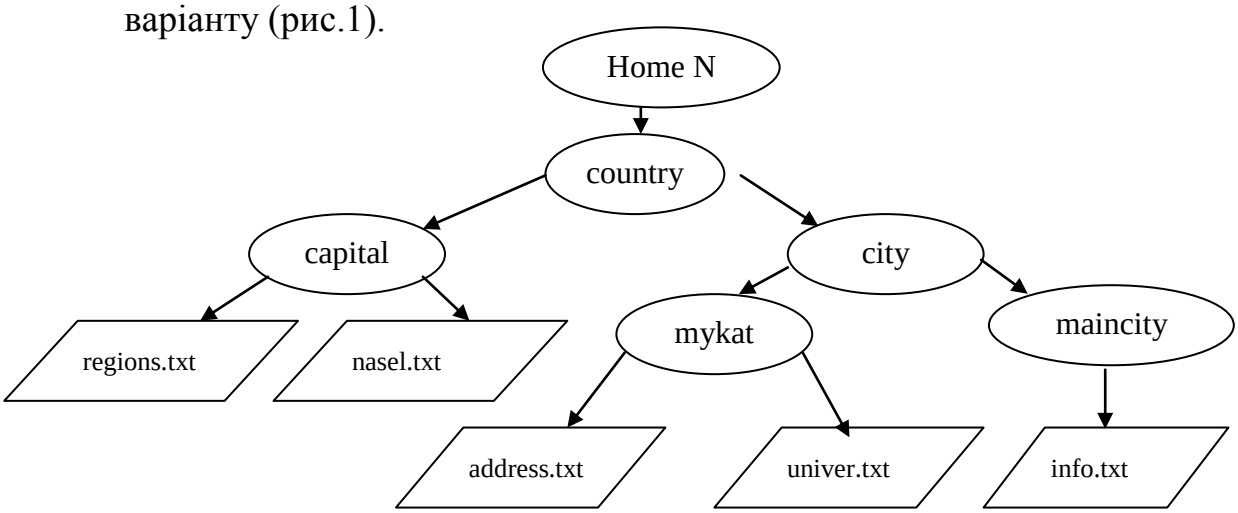

Рисунок 1 – Структура каталогів

24. Отримайте інформацію про повний шлях до каталогу country. Занотуйте шлях у звіт.
25. Скопіюйте всі файли з каталогу capital у каталог country.
    Cd /country/capital
    Cp *.* /home/country
26. У каталозі country перейменувати файл nasel.txt в nasel1.txt. Активізувати каталог country , потім mv nasel.txt nasel1.txt.
27. Ознайомитися із вмістом каталогу country.
28. Переглянути вміст файлу трьома способами –команди more, less, cat.
29. У каталозі capital створіть файл cc.txt методом злиття двох файлів regions.txt та nasel.txt
30. Утворіть жорстке посилання на файл address.txt – команда ln address.txt padres.
31. Утворіть символічне посилання на файл address.txt – ln –s address.txt sadress.
32. Перегляньте індексні дескриптори файлів та каталогів поточного каталогу city –ls –i. Переконайтесь, що файл address.txt та посилання padres мають один і той самий індексний дескриптор.
33. Виведіть на екран імена файлів у вашому каталозі, що починаються на букву *a*.
34. Знайти в каталозі country імена файлів, що закінчуються на букву s.
35. Проаналізуйте за допомогою команди history вміст роботи, продумайте відповіді на контрольні запитання.
36. Представте звіт по даній роботі викладачу.

5. Контрольні запитання

1. Що таке текстовий режим в Linux?
2. Як отримати довідкову інформацію про роботу деякої команди?
3. Які сервісні команди вам відомі?
4. Якою командою можна переглянути вміст файлу?
5. Яка команда призначена для копіювання файлу?
6. Яка команда призначена для вилучення файлу?
7. Як викликати текстовий редактор під час роботи в консолі?
8. За допомогою якої команди можна отримати детальну інформацію про файли та каталоги?
9. Як створити жорстке посилання на файл?
10. Яка відмінність між жорстким та символічним посиланням на файл?
11. Як створити символічне посилання на файл?
12. Як створити новий каталог?
13. Як вилучити каталог?
14. Як визначити об’єм вільної пам’яті?
15. Як здійснити пошук файлів в системі каталогів за фрагментами тексту?

ЛАБОРАТОРНА РОБОТА № 4

Тема: ОС Linux. Архіватори і редактори текстів

Мета: Ознайомлення з роботою в текстовому редакторі Vi. Здобуття навичок по архівації даних.

1. Принципи роботи і основні команди текстового редактора Vi

У складі ОС LINUX звичайно поставляються текстові редактори: **ed** - інтерактивний строковий редактор, **vi** і **ex** – його розширені версії. Під ім'ям **vi** (**visual interpretator** - візуальний інтерпретатор) ця програма працює як екранно-орієнтований редактор, а під ім'ям **ex** - як строчно-орієнтований. Для створення та редагування файлів призначені текстові редактори. Одним з найпоширеніших редакторів, призначених для роботи у текстовому режимі, є редактор **vi.**

Щоб викликати редактор, треба виконати команду **vi** або **vi <назва файлу>**.

Якщо такий файл уже існує, то він буде відкритий для редагування, інакше буде створено новий файл.

У найпростішому випадку для виклику редактора потрібно ввести команду **vi** text і натиснути клавішу *Enter*. На екрані з'явиться:

```
$ vi text
_
~
.
.
"text"
```

Рядок починається знаком ~, знак _ визначає положення курсору. У даний момент користувач знаходиться в командному режимі **vi**. Перейти в режим уведення тексту можна за допомогою команд додавання тексту, що не відображаються на екрані після їхнього введення:

a/A - уведення тексту після курсору/після кінця рядка (append - приєднання);

i/I - вставка тексту перед курсором/з 1-й позиції даного рядка (insert - вставити);

o/O - утворити порожній рядок нижче поточного / вище поточного.

Для виконання команд (наприклад, запису у файл, переміщення курсору) після введення тексту або його частини потрібно перейти знову в командний режим **vi**, натиснувши клавішу *Esc*. Після виклику **vi** натисніть клавішу a (уведення тексту після курсору), не натискаючи після цього клавішу *Enter*, і Ви потрапите в режим уведення тексту. Уводите текст, натискаючи клавішу *Enter* наприкінці кожного рядка (курсор у режимі введення тексту можна переміщати вправо, використовуючи клавішу "*пробіл*", і вліво, використовуючи клавішу *ВаскSpace*.

Перехід у командний режим vi. Для переходу в командний режим vi потрібно натиснути клавішу *Esc*. Тепер редактор знаходиться в командному режимі vi. У цьому режимі виконуються наступні команди:

. - повторення останньої команди;

u - анулювання дії останньої команди;

Вивчення інших численних команд цього командного режиму доцільно проводити, розбивши них на тематичні групи.

Перехід у режим ex. Щоб перейти до групи команд редактори ex (під ім'ям ex редактор працює як строчно-орієнтований), потрібно ввести символ : (двокрапка), команду і натиснути <Enter> або Esc. Команди редактора ex починаються із символу : і відображаються в нижній частині екрана. Після натискання клавіші Esc або <Enter> відбувається повернення (назад) у командний режим. Команди режиму ex:

:w - запис тексту у файл;

:r - читання файлу;

:e - редагування нового файлу;

:e! - вихід без збереження даного файлу і редагування нового;

:n - авторедагування;

:wq - запис тексту і вихід з редактора;

:x - запис тексту тільки при наявності в ньому змін;

:q! - залишити текст у робочій області і закінчити редагування;

:ab - присвоєння скорочень;

:map - визначення ключів;

:set - зміна настановних режимів;

:s - виконання заміщень.

Перехід у Shell. Редактор дозволяє в процесі роботи з ним виконувати команди ОС LINUX. Для цього потрібно перейти в командний режим Shell за допомогою команди !.

Розглянемо приклад. Визначите поточний час командою date (висновок і установка дати) :! date. Тут символ : означає перехід у командний режим ex, а символ ! дає доступ до Shell. Для тривалої роботи з командами Shell можна викликати командою :bash і після закінчення роботи повернутися в редактор vi, набравши CTRL-D.

Для повернення в командний режим vi натисніть клавішу *Enter*.

Команди, виконувані в командному режимі vi

Вивчимо групу команд режиму vi: переміщення курсору, додавання тексту, пошуку (частково), зміни і зсуви тексту, видалення, заміни букв. Команди vi не відображаються на екрані, крім команд пошуку, що починаються зі знаків / ? переміщення курсору, керування екраном дисплея, додавання тексту.

Багато команд редактора виконуються тільки при визначеному положенні курсору, і потрібно вміти користуватися клавішами керування курсором (клавіші зі стрільцями <- , -> і т.д.). Крім клавіші зі стрільцями для переміщення курсору можна використовувати клавіші: CTRL-H - уліво; CTRL-N - униз; CTRL-P - нагору; SPACE - вправо.

Команди переміщення курсору:

h - на одну позицію вліво;

l - на одну позицію вправо;

j - на одну позицію вниз;

k - на одну позицію нагору;

b - до першого символу попереднього слова;

B - те ж саме, що b, але ігноруються знаки пунктуації;

w - до першого символу наступного слова;

W - те ж саме, що w, але ігноруються знаки пунктуації;

e - до останнього символу наступного слова;

E - те ж саме, що e, але ігноруються знаки пунктуації;

(- до початку поточної пропозиції (пропозиція вважається закінченим, якщо після нього є два пробіли або порожній рядок);

) - до кінця поточної пропозиції;

{ - до початку поточного розділу (роздільником роздягнула є порожній рядок);

} - до кінця поточного розділу;

[- до початку поточної секції;

] - до кінця поточної секції;

^ - до першого відображуваного символу на поточному рядку;

O - до початку поточного рядка;

$ - до кінця поточного рядка;

H - до початку екрана;

M - на середину екрана;

L - до кінця екрана;

n - до рядка з номером n (на останній рядок, якщо номера n немає); % - до символу парної дужки, якщо курсор знаходиться під одною з них.

Команди керування екраном:

^U - зсув тексту на один рядок нагору (CTRL-U);

^D - зсув тексту на один рядок униз (CTRL-D);

^У - зсув тексту на один кадр назад (CTRL-B);

^F - зсув тексту на один кадр уперед (CTRL-F).

Щоб перемістити поточний рядок:

- у верхню частину екрана потрібно ввести команду z і натиснути клавішу *Enter*;

(у середину екрана z;

- у нижню частину екрана z- .

Для очищення екрана від повідомлень потрібно використовувати команди CTRL-R і CTRL-L; тексти в робочій області при цьому зберігаються.

Команди зміни тексту:

cw - зміна слова;

cW - те ж саме, що і cw, але ігноруються знаки пунктуації;

co - від початку поточного рядка;

c$ - до кінця поточного рядка;

cc - зміна всього рядка;

c(- від початку поточної пропозиції;

c) - до кінця поточної пропозиції;

c{ - від початку поточного розділу;

c} - до кінця поточного розділу.

Для внесення змін у текст необхідно: перемістити курсор у потрібну позицію; увести команду зміни; без пробілу набрати новий текст; натиснути клавішу ESC.

В усіх командах можна використовувати множники n, наприклад для зміни п'яти слів використовується команда з5w.

Команди пошуку починаються косою рисою / (пошук уперед по тексту) або знаком ? (пошук назад); далі випливає номер рядка або ключове слово. Команда закінчується натисканням клавіші *Enter*.

Команди зсуву тексту:

<(або>(- до початку поточної пропозиції;

<)або>) - до кінця поточної пропозиції;

<{або>{ - до початку поточного розділу;

<}або>} - до кінця поточного розділу.

У командах зсуву тексту можна використовувати множники, наприклад може використовуватися команда 2>> (зрушення вправо). Зсув установлюється командою: set sw=m. За замовчуванням m=8. Після того як курсор підведений до необхідного рядка, потрібно набрати символи << або >>.

Видалення, заміна малих літер на прописні і навпаки. Для видалення тексту/фрагмента потрібно перемістити курсор у необхідну позицію і ввести команду видалення.

dw - до кінця поточного слова;

d - те ж, що і dw, але ігноруються знаки пунктуації;

d^ - до 1-го видимого символу поточного рядка;

dO - видалення початку рядка;

d$ - видалення кінця рядка;

d(- до початку поточної пропозиції;

d) - до кінця поточної пропозиції;

d{ - до початку поточного розділу;

d} - до кінця поточного розділу;

dd - видалення всього рядка;

dkw - видалення k слів;

dk)/dk} - видалення k пропозицій, k розділів;

kdd - видалення k рядків.

Для видалення одиночного символу потрібно підвести до нього курсор і набрати x (не d), а для видалення декількох символів підряд набрати команду nx.

Для видалення тексту від початку рядка до визначеного місця і від визначеного місця до кінця рядка використовуються команди d^ і d$ відповідно.

Символ ~ використовується для заміни малих літер на прописні і навпаки. Заміна 1-й букви в останньому рядку тексту:

(Уведіть символ ((до початку поточної пропозиції).

• Наберіть команду .~

• Відновите текст командою u.

Додаткові команди:

wc [-lwc] – підраховує у файлі (за замовчуванням lwc)

-l – кількість рядків

-w – кількість слів

-c – кількість символів

sort [-rfn] – сортування у файлі

-r – обернений порядок

-f – не враховує різницю між малими і великими літерами

-n – числовий порядок сортування

u – відміна останньої зміни

U – відміна всіх змін

yw – копіювання поточного слова

yy – копіювання поточного рядка

yG – копіювання рядка від поточної позиції до кінця файлу

y$ – копіювання частини рядка від курсору до кінця рядка

y^ – копіювання частини рядка від курсору до початку рядка

p/P – після/перед курсором

r<символ> – заміна поточного символу на вказаний

<esc>:wq – вихід із збереженням файла

<esc>:q – вихід без збереженням файла

Визначення поточної робочої позиції у файлі. Після введення користувачем у командному режимі CTRL-G у нижній частині екрана з'явиться статусна інформація відповідно до положення курсору в тексті, що включає: ім'я файлу; зведення про проведену раніше модифікації; номер поточного рядка; загальне число рядків; відстань курсору від початку файлу (у відсотках).

Для закінчення роботи з редактором введіть у командному режимі :wq (запис тексту з робочої області у файл і закінчення редагування) і натисніть клавішу *Enter*. На екрані з'явиться повідомлення про те, що Ви вийшли з редактора і знаходитеся в Shell:

:wq <*Enter*>

/home/student >

2. Архіватори

В ОС Linux є декілька стандартних архіваторів: zip (підтримується також в ОС Dos), tar, cpio тощо. Тут розглядатимемо архіватор zip.

Заархівувати групу файлів можна так:

zip <повна назва архівного файлу> <файл 1> <файл 2>...<файл n>.

Наприклад, заархівувати усі файли з поточного каталогу і розмістити архів з назвою myarhiv.zip у підкаталозі /Stud можна за допомогою команди **zip/Stud/myarhiv.**

Тут тип архівного файлу зазначати не обов'язково. Розархівувати файл можна так:

unzip <назва архівного файлу>.

Щоб розархівувати файл і розташувати результати у заданому каталозі, потрібно виконати таку команду:

unzip <назва архівного файлу> -d <назва каталогу>.

Наприклад, командою **unzip /Stud/myarhiv.zip** файл myarhiv.zip буде розархівований у поточний каталог, а командою **unzip /Stud/myarhiv.zip -d /Stud/Name** файли з архіву myarhiv.zip будуть розархівовані у підкаталог Name каталогу Stud.

3. Порядок виконання роботи

1. Створити текстовий файл memo:

```
@REM AUTOEXEC.BAT DTK 386/40
ECHO OFF
Path c:\dos;c:\stacker;c:\Util;c:NC;C:\MOUSE
SET PROMT=$P$G
SET TMP=C:\TEMP
LH C:\UTIL\RKEGA
goto %config%
: student1
C:\DOS\SMARTDRV.EXE C+ 2048 1024
go to nc
: student2
APPEND E:\tc\bgi
: teacher
PATH %path%E:\windows;e:\tc\bin;e:\foxpro;
goto win
:ONC
PATH %path%G:\pctcp;
SET TZ=GMT
goto nc
:nc
nc.exe
goto end
win.com
:end
```

2. Збережіть файл та перегляньте його.

3. Редагуйте текст. Встановіть курсор на початок слова DTK в першому рядку. Перейдіть в режим вставки і наберіть 1-203. Має бути так

@REM AUTOEXEC.BAT 1-203 DTK 386/40/, натисніть Esc.

4. Поставте курсор на третій рядок і видаліть слово C:\MOUSE. Перейдіть в режим вставки і наберіть C:\GMOUSE, натисніть Esc.

5. Поставте курсор на останній рядок файла. Вставте рядок, який містить текст extention 287. Замініть слово extention на x.

6. Видаліть останній рядок.

7. Введіть команду – u

8. Встановіть курсор на п’ятий рядок, вставте перед ним пустий рядок і введіть текст: @REM 22apr.1999.

9. Підрахуйте кількість рядків, слів, символів в даному тексті.

10. Відсортуйте даний файл всіма способами

11. Збережіть зміни і вийдіть з редактору.

12. Зархівуйте всі файли каталогу capital. Архівний файл zz.zip.

13. Розархівуйте файл zz.zip

14. Заархівувати всі файли з каталогу mykat. Даний архів zz2.zip розташуйте в каталозі capital.

15. Розархівуйте файл zz2.zip.

16. Використовуючи редактор vi, написати програму на мові Сі і запустити її трансляцію в фоновому режимі.

17. Показати вихідний текст програми на мові Сі викладачу.

18. Продемонструвати виконання Сі-програми.

4. Контрольні запитання

1. Як викликати текстовий редактор в консолі?
2. За допомогою якої команди можна отримати детальну інформацію про файли та каталоги?
3. Які команди використовують для архівування та розархівування даних?
4. Яким чином можна створити архів декількох файлів?

ЛАБОРАТОРНА РОБОТА № 5

Тема: Вивчення файлової системи і функцій по обробці та управлінню даними

Мета: Вивчення структури файлової системи ОС Linux, вивчення команд створення, видалення, модифікації файлів і каталогів, функцій маніпулювання даними.

1. Теоретичні відомості

1.1. Поняття файлу

Є декілька тлумачень файлу. Логічно Файл — це деякий набір даних, які зберігаються на носіях інформації. Фізично файл — це іменована ділянка пам'яті комп'ютера, призначена для зберігання даних. Усі об'єкти ОС Linix є файлами. Наприклад, файлами є текстові документи, програми, електронні таблиці, каталоги, а також пристрої вводу-виводу і навіть сама ОС теж є файлом. Одні файли ми лише використовуємо, інші — створюємо.

Файл має такі властивості: назву, тип, дату створення, розмір та адресу.

Назву файлу дає користувач, її можна змінювати. У ОС Linux розрізняють великі та маленькі літери в назвах об'єктів. Наприклад, MyFile.txt та myfile.txt — це назви різних файлів. Тип вказує на те, якою програмою слід опрацьовувати певний файл. Назву файлу від назви типу розмежовують крапкою. Отже, ім'я файлу може мати такий вигляд: назва файлу.назва типу. Наприклад, auth.log, syslog.conf, fileS.tar.

Одним із важливих різновидів файлів є так звані **спеціальні** (логічні) файли. Це файли, що відповідають стандартним системним (зовнішнім) пристроям. Зокрема, клавіатура, миша чи монітор трактуються як спеціальні файли. Принцип роботи системи полягає в тому, що вона звертається до спеціального файлу, зчитує інформацію з нього і передає керування драйверу відповідного пристрою - програмі, яка забезпечує обмін командами та даними між процесором і цим пристроєм. Імена деяких файлів (без типу) стандартних пристроїв наведені у табл. 1.

Таблиця 1. Файли стандартних пристроїв

Назва файлу	Пристрій
ttyN	Консоль N, де N - це число
mouse	Миша
audio	Звукова плата
modem	Модем
ttySN	Послідовний порт з номером N
IpN	Паралельний порт з номером N
hdxN	Жорсткий диск N
fdO	Перший дисковід для гнучких дисків
null	Порожній пристрій

1.2. Файлова структура

Ще одним різновидом файлів є каталоги. Каталог — це файл, у якому записана інформація про файли та інші каталоги, які містяться у ньому. Каталог, що міститься у іншому каталозі, називається підкаталогом цього каталогу. Підкаталог теж може містити інші каталоги. Назви підкаталогів відокремлюються один від одного похилою рискою "/".

Кореневим каталогом (каталогом найвищого рівня) є каталог /root (інколи позначають просто /). Усі інші каталоги є підкаталогами кореневого. Каталогами першого рівня є системні стандартні каталоги. Деякі назви системних каталогів та їхнє призначення наведемо у табл. 2.

Таблиця 2. Системні каталоги

/	Кореневий каталог.
/root	Каталог адміністратора системи. Доступ інших користувачів захищений паролем.
/home	Каталог, який містить підкаталоги всіх користувачів системи.
/boot	Каталог, який містить файли необхідні для завантаження системи, а також ядро Linux.
/etc	Каталог, який містить основні конфігураційні файли, а також ті файли, які вимагаються для початкової стадії завантаження ОС.
/lib	Каталог, який містить так звані бібліотеки, необхідні компілятору C(C++).
/mnt	Каталог, який містить всі змонтовані в Linux пристрої(які мають свою файлову систему та містять інформацію).
/tmp	Папка для тимчасових файлів.
/lost+found	Каталог, де складаються загублені в результаті зовнішньої ситуації файли. Сюди звертається програма відновлення системи.
/dev	Каталог файлів пристроїв.
/usr	Найбільший каталог, в ньому заходяться всі встановлені в системі програми і додатки.
/bin	Інформація про всі команди

На комп'ютері можна встановити відразу декілька ОС, Іаприклад, Linux, Windows, OS/2 тощо. Кожній ОС надається своя ділянка на твердому диску, яка називається розділом. 5ауважимо, що в Linux є можливість не тільки переглядати файли інших ОС, але й опрацьовувати їх. Для цього відповідні розділи інших ОС потрібно змонтувати (під'єднати) у системному каталозі /mnt. Це виконують за допомогою команди mount, яка описана у додатку. У цей самий каталог монтують (під'єднують) розділи для роботи з дискетою (/mnt/floppy) та CD-диском (/mnt/cdrom). Роботу з дискетою та CD-диском розглянемо пізніше.

Зазвичай вміст кореневого каталогу доступний користувачеві лише для перегляду. Власні файли та каталоги користувач повинен створювати у каталозі

/home/<ім'я користувача> (домашній каталог) або в його підкаталогах. З огляду на це виникають поняття шляху до файлу (адреси) та повної назви файлу. Шляхом до файлу (адресою) називаються записані через (/) назви каталогів від деякого заданого, наприклад, /home, до каталогу, що містить цей файл. Наприклад, нехай на деякому комп'ютері у домашньому каталозі є каталог Stud, а в ньому — файл з (короткою) назвою lab.txt. Тоді адреса файлу lab.txt така: /home/Stud/, а повна назва файлу lab.txt -/home/Stud/lab.txt. Останню похилу риску в кінці назви каталогу ставити не обов'язково. Тут вона означає, що Stud є каталогом.

Зверніть увагу на те, що каталоги в ОС Linux трактуються як файли, а також на те, що логічні диски не є об'єктами файлової системи, як в ОС Windows.

Над файлами визначені такі дії: створення, вилучення, копіювання, переміщення, перейменування тощо. Крім цього, саме в ОС Linux прийнято визначати права доступу для файлів, тобто робити їх доступними лише для читання, лише для записування, читання чи записування, лише для виконання певним користувачем чи членами групи. Усі ці дії можна реалізувати за допомогою команд з контекстного меню, а права доступу регулюються командою Властивості.

Зауважимо, що адміністраторові root завжди доступні файли усіх користувачів системи.

1.3. Права доступу до файлів і каталогів та керування ними

chmod – зміна атрибутів захисту файла.

Кожен файл чи каталог має власний набір атрибутів щодо прав доступу до нього. В ОС Linux розрізняють **три основні рівні доступу** до файлів:

1) доступ власника файлу чи каталогу (u – атрибут власника)
2) доступ групи, до якої належить власник (g – атрибут для групи)
3) усі інші користувачі (o – атрибут для інших користувачів)
   для всіх категорій користувачів (a – атрибут для всіх)

Для кожного рівня існують свої **способи доступу** до файлу та каталогу:

– читання (r)
– записування (w)
– виконання (x)
– відсутність доступу (-)

Дозвіл на читання файлу означає, що вміст файлу можна переглядати, а дозвіл на читання – що вміст можна переглядати і редагувати. Дозвіл на виконання означає, що файл можна запускати на виконання. Для каталогу дія читання означає, що вміст його можна переглядати. Записування – у ньому можна створювати та вилучати підкаталоги та файли, виконання – стають доступними всі атрибути прав доступу для підкаталогів чи файлів, які в ньому розміщені.

Окрім читання, записування та виконання, для файлів та каталогів можна визначити спеціальні права доступу SUID(Set User ID root) та SGID(Set Group

ID root). Ці права дають змогу певним користувачам або групі використовувати файл чи каталог на праві адміністратора root.

Щоб з'ясувати усі атрибути файлів та каталогів, треба застосувати команду ls –l <назва каталогу>. Наприклад:

1) -rw-r—r--

перша - – вказує на файл
(rw-) – права доступу для власника файлу(читання та записування)
(r--) – права доступу для групи власників (тільки читання)
(r--) – права доступу всіх інших (тільки читання)

2) -rwx------

(rwx) – права доступу тільки для власника файлу (читання, записування, виконання)

3) drwxr-x--x

d – вказує на каталог
(rwx) – права доступу тільки для власника файлу(читання, записування, виконання)
(r-x) – права доступу для групи власників (тільки читання і виконання)
(--x) – права доступу всіх інших (тільки виконання)

-	звичайний файл
d	каталог
l	Символічне посилання
c	Символічний пристрій

Виконувана операція кодується за допомогою таких символів :
= – встановлює значення всіх атрибутів для даної категорії користувача
+ – додає атрибут для даної категорії
- – відміняє атрибут для даної категорії

Наприклад:

1) Дозволити доступ до читання і запису до файлів з іменем mar власнику і групі власників

chmod ug+rw mar.*

2) Встановити власнику файла may права на виконання

chmod u+x may

3) Лишити власника файла may прав доступу на виконання

chmod u-x may

4) Відмінити права на читання каталогу mountly для групи і всіх інших

chmod g-r o-r mountly

Зауваження

Права доступу можуть бути задані в команді не тільки в символьному виді, але і в цифровому(вісімкове число). Зв'язок між цифровою і символьною формою наведена в таблиці 3.

Таблиця 3

Цифрова форма		Символьна форма
двійкова	вісімкова	
111	7	rwx
110	6	rw-
101	5	r-x
100	4	r--
011	3	-wx
010	2	-w-
001	1	--x
000	0	---

Наприклад, встановити атрибути для читання і запису для власника і групи і тільки для читання для інших

-rw-rw-r—

110110100_2

664_8 .

2. Порядок виконання роботи

1. Виведіть інформацію про вміст каталогу capital за допомогою команди ls. Запишіть дану інформацію у звіт.

2.Передивіться вміст файлу info.txt

1. Скасуйте право доступу для читання файлу (nasel.txt)для інших користувачів.

2. Скасуйте право для запису файлу (regions.txt)для всіх користувачів.

3. Виведіть інформацію про вміст каталогу capital за допомогою команди ls –l і запишіть зміни.

4. Надайте право доступу для запису власникові ті групі файлу regions.txt

5. Скасуйте право для читання всіх користувачів файлу info.txt. Перевірте правильність введеної команди(спробуйте команду more info.txt).

6. Надайте право доступу для читання власнику та групі файлу info.txt. Занотуйте зміни. Знову спробуйте відкрити файл.

9. Використовуючи команди ОС LINUX, створити два текстових файли.

10. Отримані файли об'єднати в один файл і його вміст переглянути на екрані.

11. Створити нову директорію і перемістити в неї отримані файли.

12. Вивести повну інформацію про усі файли і проаналізувати рівні доступу.

13. Додати для всіх трьох файлів право виконання членам групи й інших користувачів.

14. Переглянути атрибути файлів.

15. Створити ще один каталог.

16. Установити додатковий зв'язок об'єднаного файлу з новим каталогом, але під іншим ім'ям.

17. Створити символічний зв'язок.

18. Зробити поточний новий каталог і вивести на екран розширений список інформації про його файли.

19. Зробити пошук заданої послідовності символів у файлах поточної директорії й одержати перелік відповідних файлів.

3. Контрольні запитання

5. Що називається файлом в ОС Linux?
6. Поясніть зв’язки між файлами і способи їх створення.
7. Що визначає атрибути файлів і яким способом можна їх переглянути?
8. За допомогою якої команди можна змінити права доступу до файлу чи каталогу?
9. Які способи доступу до файлів чи каталогів ви знаєте?
10. Як визначити право доступу на читання для членів групи?
11. Які права надасть команда chmod g-r file.txt?
12. Які спеціальні права доступу вам відомі?

ЛАБОРАТОРНА РОБОТА №6

Тема: Основи shell-програмування

Мета: Вивчити основні команди мови shell та вивчити можливості створення shell-процедур.

1. Теоретичні відомості

1.1. Інтерпретатор shell

Інтерпретатор shell виконує інтерфейсні функції між ОС і користувачем. Він перехоплює всі команди користувача: формує і виводить відповідні повідомлення. Крім запуску на виконання стандартних команд у виконуваних файлах, інтерпретатор містить свою власну мову, яка за своїми можливостями наближається до мов програмування високого рівня.

Оператори мови дозволяють створювати команди. Такі програми а також командні файли, які містять їх називають **shell-процедурами** або **shell-файлами**. Від звичайних програм вони відрізняються засобами обробки. Процедура не вимагає компіляції. Вона аналогічна до командного рядка MS-DOS, але з більшими можливостями. Текст процедури набирається як звичайний текстовий файл. Для цього доцільно використовувати редактор Vi. Перевірений і відлагоджений shell-файл може бути запущений на виконання наступним чином:

1) $ chmod u+x shfil
   $shfil
   $

Така форма припускає, що файл новий і його потрібно зробити виконуваним.

2) $ sh -c “shfil” або
   $sh shfil
   $

За командою sh викликається другорядний інтерпретатор shell, і в якості аргументу йому передається командний рядок, який містить ім’я файла даної процедури shfil, що знаходиться в поточному каталозі Процедурі shell можуть бути передані аргументи при запуску, кожному з яких ставиться у відповідність позиційний номер: від $1 до $9 ($0 – ім’я самої процедури). До кожного із 10 аргументів можна звернутись вказавши номер його позиції в процедурі.

1.2. Переключення вводу-виводу інтерпретатора

>файл	Перенаправлення стандартного вихідного потоку у файл
>>файл	Додавання стандартного вихідного потоку у файл
<файл	Отримання стандартного вихідного потоку із файлу
p1|p2	Передача вихідного потоку програми p1 в якості вхідного потоку для програми p2
n>файл	Перенаправлення вхідного потоку із файлу з дескриптором n у файл
n>>файл	Додавання записів вихідного потоку із файлу з дескриптором n у файл
n>&m	Злиття потоків з дескриптором n та m

1.3. Додаткові команди для роботи з текстовими файлами

1) **grep** –пошук шаблону(підрядка)у файлах.

$grep [-ключі] підрядок список_файлів.

-c – вивід імен всіх файлів з вказаною кількістю рядків, які містять шаблон; -i – ігнорування регістра; -n – вивід перед рядком його відносного номера; -v – вивід рядків, що не містять шаблони; -l – вивід тільки імен файлів, які містять шаблон.

2) **cmp** – вивід місця першого входження.

$cmp файл_1 файл_2.

3) **diff** – вивід всіх розходжень у файлах.

$diff файл_1 файл_2.

4) **find** – пошук файлів у під дереві каталогів.

$find список_каталогів умови_пошуку.

find / -name file1 — знайти файли і директорії з іменем file1. Пошук почати з кореня (/);

find / -user user1 — знайти файли і директорії належним user1. Пошук почати з кореня (/);

find /home/user1 -name "*.bin" — знайти всі файли і директорії, імена яких закінчуються на '. bin'. Пошук почати з '/ home/user1'*;

find /usr/bin -type f -atime +100 — знайти всі файли в '/usr/bin', час звернення до яких більше 100 днів;

find /usr/bin -type f -mtime -10 — знайти всі файли в '/usr/bin', створених або змінених протягом останніх 10 днів;

find / -name *.rpm -exec chmod 755 '{}' \; — знайти всі файли і директорії, імена яких закінчуються на '.rpm', і змінити права доступу до них;

find / -xdev -name "*.rpm" — знайти всі файли і директорії, імена яких закінчуються на '.rpm', ігноруються носії cdrom, floppy і т.д.

locate "*.ps" — найти всі файли які містять в імені '.ps'. Попередньо рекомендується виконати команду 'updatedb'.

whereis halt — показує розміщення бінарних файлів, вихідних кодів, які відносяться до файлу 'halt'.

which halt — відображає повний шлях до файлу 'halt'.

1.4. Деякі команди shell

1) **echo** – вивід повідомлень.

a) $echo hello world
hello world
$

b) $echo *
перерахує всі файли поточного каталогу
$

c) $shfil p1 p2 p3
$echo $3
p3
$
#процедура shfil із трьома параметрами

2) **banner** – вивід повідомлень у верхньому регістрі.

$banner ‘hello’

HELLO
$

3) **read** – читання

$read greeting
hello world
$echo $ greeting
hello world
$

4) **set** – присвоєння значень позиційним параметрам.

Приклад 1

$set a1 ab2. abc
$echo $1 $2
a1 ab2$
$

shift – змістити позиційні параметри вліво на одну позицію. Після виконання shift попереднє значення параметра $1 зникає, значення $1 набирає значення $2, значення $2 - значення $3 и т.д.

Продовження попереднього прикладу:

$shift
$echo $1 $2
ab2 abc
$

5) **test** – перевірка умови (використовується в умовному операторі, повертає 1 – true, 0 – false).

a) Перевірка файлів.

test - ключ ім'я_файла

Ключі: -r – файл існує і доступний для читання; -w – файл існує і доступний для запису; -x – файл існує і доступний для виконання; -f – файл існує і має тип '-', тобто звичайний; -s – файл існує і має тип '-', тобто непустий; -d – файл існує і має тип 'd ', тобто каталог.

b) Порівняння числових рядків.

test число_1 -ключ число_2

-ed – рівно; -ne – не рівно; -lt – менше; -le – менше або рівно; -gt – більше; -ge – більше або рівно.

Приклад 2

$x=5
$[$x -lt 7]
$echo $?
0
$[$x -gt 7]
$echo $?
1
$

c) Порівняння рядків.

test [-n] рядок – рядок непустий; test –z рядок – рядок пустий;

test рядок1=рядок2 – рядки рівні; test рядок1!=рядок2 – рядки не рівні.

Приклад 3

```
$x  =  abc
$["$x"  =  "abc"]
$echo $?
0
$["$x"  !=  "abc"]
$echo $?
1
$
```

Особливості порівняння чисел і рядків. Shell трактує всі аргументи як числа в випадку, якщо здійснюється порівняння чисел, і всі аргументи як рядки, якщо здійснюється порівняння рядків.

Приклад 4

```
$X = 03
$Y =3
$["$X  -eq  "$Y"]          - порівнюються значення чисел
$echo $?
0
$["$X  =  "$Y"]         - числа порівнюються як рядки символів
$echo $?
1
$
```

1.5. Змінні мови shell

Мова shell дозволяє працювати із змінними без попереднього оголошення. Імена починаються із букви і можуть містити букви і цифри. Звернення до імен починається із знаку $.

Перехід до початкового каталогу: $cd $HOME

Вбудовані змінні інтерпретатора

$#	Кількість аргументів
$*	Всі аргументи, які передають інтерпретатору
$@	Аналогічно $*
$-	Прапорці, які передають інтерпретатору
$?	Повертає значення останньої виконаної команди
$!	Номер процесу останньої команди, яка запущена за допомогою &
$$	Номер процесу інтерпретатора
$HOME	Аргумент, прийнятий за замовчуванням для команди cd
$IFS	Список символів, які розділяють слова в аргументах
$MAIL	Файл, зміна якого приводить до появи повідомлення “you have a mail”
$PATH	Список каталогів, в яких здійснюється пошук команд
$PS1	Рядок – запрошення, за замовчуванням $
ffps-	Рядок – запрошення при продовженні командного рядка, за замовчуванням >

1.6. Оператори мови shell

1) ***Оператор присвоєння*** здійснюється за допомогою „=” без пропусків(c=3).

Основні команди здійснюються за допомогою команди **expr** і операторів.

Арифметичні		Логічні	
+	Додавання	=	рівно
-	Віднімання	!=	Не рівно
*	Множення	\<	менше
/	Ділення	\<=	менше або рівно
%	Остача від ділення	\>	більше
		\>=	більше або рівно

Результат логічних операцій 1=true 0=false. Всі операції та операнди є самостійними аргументами команди expr і тому повинні відділятися один від одного і від команди expr пропусками.

Приклад 5

```
# Текст процедури:
$a=2
$a=`expr $a + 7`
$b=`expr $a / 3`
$c=`expr $a - 1 + $b`
$d=`expr $c % 5`
$e=`expr $d - $b`
$echo $a $b $c $d $e
# Результат роботи процедури:
 9 3 11 1 -2
$
```

2) ***Умовний оператор***

```
if список_команд1
    then список_команд2
    else список_команд3
fi
```

Приклад

```
if test=”$1”=hello
then
        command
fi
```

Для задання пустого списку команд використовують спеціальний оператор „**:”**

3) ***Оператор вибору***

```
case слово in
   шаблон)команди;;
   шаблон)команди;;
   .....
esac
```

Приклад

```
case ”$1” in
hello)command;;
esac
```

4) ***Цикли***

а)
```
while команди1
список_знач
   do
    команди2
  done
```

б)
```
until команди1
do
   команди2
done
```

в)
```
for змінна in
do
      команди
done
```

Приклад 6

Ввід рядка із декількох слів. Підрахунок і вивід кількості символів у кожному слові.

Текст процедури, якій присвоєно ім'я P1:

```
echo “Input string:”
read A
set  $A
while [ “$1” ]
do
   echo “$1 = `expr ”$1” : ‘.*’`”
   shift
done
```

Результат роботи процедури:

```
$P1
Input string:
df ghghhhh aqw
df = 2
ghghhhh = 7
aqw = 3
$
```

Приклад 7

Вивести на екран слово(поле), номер якого(змінні N) вказана в параметрі при звернені до процедури, ім'я якої P2. процедура запрошує ввід рядка з клавіатури. Номер слова задається як аргумент процедури.

Текст процедури P2:

```
 i=1  - лічильник номерів слова в рядку, формується при кожному
 виконанні циклу
   N=$1                          значення першого параметру
   echo "Введіть рядок: "
   read a
   set  $a
   while test $i -lt $N
    do
     i=`expr $i + 1`              - формування номеру наступного слова
     shift
    done
   echo "$N поле строки: \"$1\""
```

Приклад роботи процедури P2:

```
$P2 2                    <NewLine>              - N=2
Введіть рядок:  aa bb cc dd      <NewLine>
2 поле рядок: "bb"
$
```

Приклад 8

Текст процедури, яка виводить в стовпець список імен файлів поточного каталогу.

```
list =`ls`
for val in $list
do
  echo "$val"
done
echo end
```

При розв’язанні логічних задач, зв’язаних з **обробкою символьних рядків (текстів)** команда expr може бути використана, наприклад, як засіб для підрахунку символів в рядках або для виділення із рядка ланцюжка символів.

Операція обробки рядків задається **кодом операції** "**:**" і шаблонами.

'.*' - шаблон для підрахунку кількості символів в рядку,

'...\(.*\)....' - шаблон для виділення підрядка видаленням символів рядка, які відповідають точкам в шаблоні.

Приклад 9

```
$ m=aaaaaa
$ expr  $m : '.*'
6
$
```

Приклад 10

```
$ n=abcdefgh
$ expr  $n : ✧...\(.*\)..'
def
$
```

1.7. Bash процедури

Приклад простої процедури

```
#!/bin/bash
echo -e "Hi, please type the word: \c "
read word
echo "The word you entered is: $word"
echo -e "Can you please enter two words? "
read word1 word2
echo "Here is your input: \"$word1\" \"$word2\""
```

1.8. Масиви

У BASH реалізовані тільки одномірні масиви. Індексами можуть бути як цілочисельні значення (в цьому випадку масив являє собою вектор), так і рядки (асоціативний масив або «хеш»).

Незважаючи на уніфікацію синтаксису для роботи з векторами і асоціативними масивами, перші не є різновидом других ні з точки зору кінцевого користувача, ні з точки зору внутрішньої логіки самого інтерпретатора.

Загальні операції:

1. Отримати рядок який містить всі елементи масиву - values=${arr[@]}
2. Здійснити масову заміну першого входження підрядка «MAY» на

«MARCH» у всіх елементах масиву і повернути рядок, складену з результуючих елементів масиву, розділених пропуском: values=${arr[@]/MAY/MARCH}.

3. Теж саме що і в попередньому пункті але будуть замінені всі входження підрядка : values=${arr[@]//MAY/MARCH}

Операції з індексованими масивами:

Bash має підтримку одновимірних масивів. Ініціалізувати елементи масива можна наступним чином my_array[xx]. Також можна явно оголосити масив в скрипті через директиву declare: declare –a my_array. Звертання до елементів масиву: ${my_array[xx]}.

1. Ініціалізація масиву :
   Array=(element1, element2, element3)
   tmp_arr[0]=elemnt1
   tmp_arr[0]=elemnt2
   tmp_arr[0]=elemntN
2. Додавання елментів в кінець масиву
   arr+=(newe_lement)
3. Отримати перший елемент масиву
   Echo “${arr[0]}”
4. Отримати останній елемент масиву
   Echo “${arr[@]:${#arr[@]}-1}”

2. Варіанти завдань

1. Розробіть текст процедури з використанням vi за завданням, варіант завдання призначається викладачем.

2. Налагодьте, при необхідності відредагуйте й виконаєте процедуру.

3. Оформіть процедуру з використанням допоміжних команд і коментарів так, щоб вона легко читалася й щоб результати її роботи легко аналізувалися.

Завдання першого рівня

Розробити shell-процедуру з коментарями, що виконує нижче перераховані функції.

1. Процедурі передаються значення N і M у якості параметрів, після чого від процедури поступає запит на введення послідовності слів(слова розділені пробілом). Процедура аналізує послідовність N перших слів і виводить ті слова, довжина яких більше M.

2. Вводить рядок із заданої кількості слів. Виділяє слова, що починаються на зазначену параметром букву, підраховує число таких слів.

3. Вводить рядок з N слів, аналізує довжину кожного слова, упорядковує слова за алфавітом й виводить список на екран. Значення N задається параметром.

4. Виводить запит на ввід рядка слів, після чого виводить N слів супроводжуючи кожне слово порядковим номером. Значення N задається параметром.

5. Вводить довільну кількість коротких символьних параметрів, підраховує довжину кожного з них і виводить на екран список значень довжин і загальну кількість введених параметрів.

6. Вводить кілька коротких чисел у вигляді параметрів, підраховує їхню суму й результат виводить на екран.

7. Запрошує ввід послідовно декількох чисел зі знаками й виводить на екран два списки чисел - позитивних і негативних.

8. Здійснює пошук в особистому головному каталозі користувача створені ним файли, виводить список їхніх імен і роздруковує текст файлу, заданого користувачем.

9. Створює новий підкаталог і поміщає туди нові файли, створювані користувачем по запитах процедури. Імена нових файлів вказуються параметрами.

10. Створює новий підкаталог і копіює туди з батьківського каталогу файли заданого параметром типу.

11. Аналізує зазначений параметром каталог і виводить на екран число файлів різного типу (звичайні, директорії, сховані). Тип задається параметром.

Завдання другого рівня

12. Написати shell-процедуру що:

- уводить передану, в якості 1-го параметра, кількість символьних рядків;
- у кожному уведеному рядку шукає підрядок, переданий як другий параметр;
- заміняє кожен знайдений підрядок на рядок, переданий як третій параметр;
- виводить на екран уведений рядок і відповідний йому змінений рядок

13. Написати shell-процедуру що:

- уводить 2 символьні рядки;
- у кожному уведеному рядку шукає підрядок, переданий як другий параметр;
- заміняє кожен знайдений підрядок на пробіл;
- утворює із отриманих рядків третій рядок так, щоб у ньому чергувалися слова з першого й другого рядків;
- виводить на екран введені рядки і новий рядок.

14. Написати shell-процедуру що:

- уводить символьний рядок;
- у введеному рядку шукає підрядок, переданий як перший параметр;
- вставляє після кожного знайденого підрядка символ, переданий як другий параметр;
- видаляє з отриманого рядка символ, переданий як, третій параметр;
- виводить на екран введений рядок і новий рядок.

15. Написати shell-процедуру що:

- уводить символьний рядок;
- перевіряє уведений рядок на співпадіння з рядком, що переданий як перший параметр;

- якщо рядки співпадають, то видає на екран запрошення повторити введення;

- інакше, то порівнює довжину уведеного рядка з довжиною 2-го параметра і, якщо вони рівні, виводить на екран уведений рядок у зворотному порядку.

16. Написати shell-процедуру що:

- уводить символьний рядок;

- перевіряє уведений рядок на спів падіння з рядками, що містяться у файлі, ім'я якого передається в якості 1-го параметра;

- для всіх знайдених співпадінь заміняє відповідні рядки у файлі на рядок переданий у якості 2-го параметра;

- виводить на екран старий і новий вміст файл, а також число знайдених співпадінь.

17. Написати shell процедуру що:

- уводить символьний рядок, що містить маршрутне імя деякого файлу, перевіряє уведене маршрутне ім'я, якщо воно починається з символу /, на збіг його першої частини з маршрутним іменем домашнього каталогу користувача;

- якщо уведене маршрутне ім'я містить маршрутне ім'я домашнього каталогу або є відносним, то перевіряє існування зазначеного файлу, у противному випадку виводить на екран повідомлення про помилку;

- якщо файл існує, то виводить на екран його вміст;

- якщо файл не існує, то створює його й записує в нього рядок, переданий як параметр.

18. Написати shell-процедуру що:

- уводить символьний рядок, що містить ім'я деякого файлу;

- перевіряє наявність файлу в домашньому каталозі або в одному з підкаталогів;

- якщо файл існує, то виводить на екран його вміст;

- інакше створює його і записує в нього з консолі деякий текст;

- установлює для файлу права доступу передані як параметр.

19. Написати shell-процедуру що:

- віддаляє із заданого першим параметром каталогу й всіх підкаталогів файли, дата останньої модифікації яких передує поточній даті мінус число днів, передане як другий параметр;

- змінює дату модифікації всіх інших файлів зазначеного каталогу на поточну буз зміни вмісту файлів

20. Написати shell-процедуру що:

- виводить на екран список всіх користувачів системи, включених у задану першим параметром групу користувачів;

- для кожного із заданих третім і наступними параметрами імен користувачів виводить на екран права доступу до заданого другим параметром файлу.

21. Написати shell-процедуру що:

- у заданому першим параметром каталозі знаходить всі прості файли, число посилань на які максимальне, і вилучає їх;

- вилучає всі порожні каталоги;

- видає на екран повідомлення про кожен вилучений файл і каталог.

22. Написати shell-процедуру що:

- в заданому першим параметром каталозі знаходить всі підкаталоги число простих файлів у яких більше заданого другим параметром числа;

- видаляє знайдені підкаталоги;

видає на екран повідомлення про кожен вилучений підкаталог.

23. Написати shell-процедуру що:

- в заданому першим параметром каталозі знаходить всі прості файли, у яких міститься заданий другим параметром символьний рядок;

- у знайдених файлах видаляє всі повторювані рядки;

- виводить на екран імена всіх знайдених файлів.

24. Написати shell-процедуру що:

- у каталозі ім’я якого передається першим параметром, знаходить всі прості файли розміром більше заданого другим параметром;

- створює в зазначеному каталозі 3 нові підкаталоги;

- поміщає у створені підкаталоги файли з вихідного каталогу: у перший – файли, що містять один рядок із заданим словом, у другий – файли із двома такими рядками, у третій – з трьома;

- імена всіх файлів не включених у нові підкаталоги, виводить на екран.

25. Написати shell-процедуру що:

- обчислює значення арифметичного виразу, заданого першими 7-ма параметрами;

- порівнює отримане значення із числом, що вводиться в процедуру;

- при збігу результатів порівняння, виводиться на екран заданий вираз і його значення.

26. Написати shell-процедуру що:

- визначає висоту піддерева каталогів, починаючи від каталогу, переданого як перший параметр;

- виводить на екран повне маршрутне ім’я каталогу, останнього в структурі піддерева максимальної довжини.

27. Написати shell-процедуру що:

- читає вміст першого файлу, переданого як перший параметр;

- читає вміст другого файлу, переданого як другий параметр;

- виводить на екран кожні 7 секунд поперемінно 2 рядки з першого й 1 рядок із другого файлу, переміщаючись по файлах циклічно.

28. Написати shell-процедуру що:

- вводить символьний рядок що містить деяке ціле число, здійснити перевірку введеного значення, якщо введено не число, то вивести повідомлення;

- читає вміст файлу переданого як перший параметр;

- виводить на екран кожні 6 секунд 2 рядки з першого файлу і 1 введений рядок.

29. Написати shell-процедуру що:

- вводить символьний рядок, що містить два цілих числа m та n розділених пробілом;

- читає вміст файлу, переданого як перший параметр;

- виводить на екран кожні 5 секунд m рядків з файлу та n рядків «Будь здоровий».

3. Контрольні запитання

1. Що таке shell-процедура? Призначення.

2. Якого типу команди можуть бути включені в тіло процедури?

3. Чим відрізняється обробка процедури при виконанні від обробки програми мовою високого рівня?

4. Що таке параметри? Для яких цілей вони використаються? Яке число параметрів може бути передано процедурі?

5. Які допоміжні команди Ви використали при оформленні процедури?

6. Якого виду значення і як можуть бути привласнені змінної мови shell?

7. Як здійснювати розгалуження обчислювального процесу процедури?

8. Якого типу цикли в процедурах можуть бути побудовані засобами мови shell?

9. Які способи виклику процедур на виконання Ви знаєте?

ЛАБОРАТОРНА РОБОТА №7

Тема: Управління процесами

Мета: Вивчити основні можливості роботи з процесами ОС Windows та Linux.

1. Управління процесами і потоками в ОС Windows

Перегляд і управління процесами під Windows виконують за допомогою утиліти „Диспетчер задач”. Після запуску вона має такий вигляд

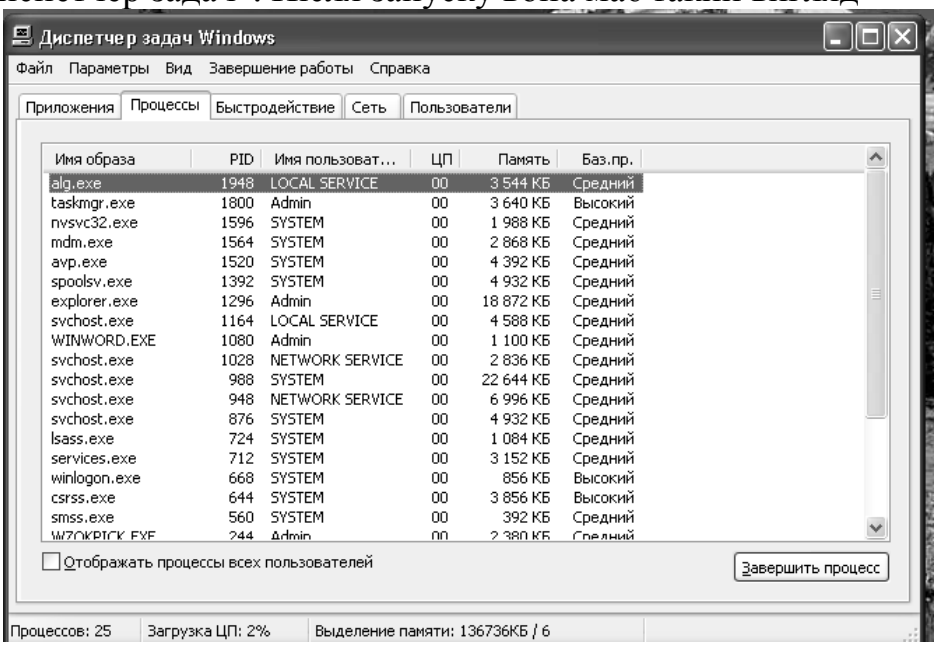

Рисунок 1

Тут відображається інформація про програми і процеси, які виконуються на комп’ютері. Диспетчер задач служить для відображення ключових показників швидкодії комп’ютера. Для виконуваних програм можна переглянути їх стан і закінчити програми, які перестали відповідати на запити. Є можливість перегляду активності виконуючих процесів з використанням до 15 параметрів, а також графіків і інформації про використання центрального процесора і пам’яті. Можна сортувати процеси за будь-яким критерієм, клацнувши мишкою на відповідній колонці. Клацнувши мишкою по вибраному процесу, за допомогою контекстного меню ви можете змінити базовий пріоритет процесу, примусово закінчити процес і переглянути додаткові параметри.

Запустивши утиліту, запустіть декілька додатків (наприклад: Word, Paint, Exsel і т.д), зверніть увагу на зміни у вікні процесів.

Завдання 1

Згідно номеру варіанту виконайте наступні дії (табл.1). Відсортуйте процеси за заданим критерієм. Опишіть один із системних процесів. Запустіть зазначений додаток. Опишіть процес , який виник за заданими характеристиками. Примусово завершіть зазначений процес. Виконувані дії ілюструйте копіями екранів.

Таблиця 1

№	Критерій	Додаток	Характеристики
1	Показати дерево системних процесів	Exsel	Переглянути додаткові. властивості процесу
2	Відсортувати по PID	Блокнот	Переглянути додаткові. властивості процесу
3	Відсортувати по завантаженню процесора	Wordpad	Змінити пріоритет користувацького процесу
4	Відсортувати по пріоритету	Калькулятор	Переглянути додаткові. властивості процесу
5	Відсортувати по власнику	Paint	Змінити пріоритет користувацького процесу
6	Показати дерево користувацьких процесів	Провідник	Переглянути додаткові властивості процесу
7	Відсортувати по найменуванню	Редактор реєстру	Змінити пріоритет користувацького процесу
8	Відсортувати по пріоритеті	Web-браузер	Змінити пріоритет користувацького процесу
9	Відсортувати по завантаженню процесора	Мережне оточення	Переглянути додаткові властивості процесу
10	Показати дерево користувацьких процесів	Дефрагментація диску	Переглянути додаткові властивості процесу

Копії екрану виконаним завданням і опис виконаних дій записати у звіт.

2. Управління процесами і потоками в ОС Unix/Linux

2.1. Керування процесами в UNIX /Linux

UNIX - багатокористувацька та багатозадачна операційна система. Концепція процесу є базовою для архітектури ОС. Процес будується для кожної прикладної (наприклад sh-процедури і системної задачі (наприклад - утиліти) за допомогою системного виклику fork. Він є одиницею обчислювальної роботи і споживання ресурсів. У процесі життя в системі, процес безпосередньо керується спеціальними системними викликами, що забезпечують його побудову - передачу керування - завершення:

fork - exec - exit сукупність цих переключень станів процесу визначають час існування процесу.

У загальному випадку активний процес (виконувана задача) може знаходитися в одному із шести станів (у Linux):

- очікування процесора в черзі
- виконання на процесорі протягом виділеного кванта часу

- очікування звільнення ресурсу (наприклад – пристрою)
- припинений спеціальним сигналом
- завершився, але його дескриптор ще в оперативній пам'яті ядра ОС
- процес „свопірований” у зовнішню пам'ять .

В ОП в активному стані знаходиться кілька процесів - до 30000. .Кожному процесові виділяється віртуальна пам'ять практично без обмежень (за рахунок сторінкової організації).

При виконанні будь-якого обчислювального завдання прикладні і системні процеси ядра повинні взаємодіяти між собою. Це необхідно для обміну даними, для передачі керуючої інформації при використанні системних ресурсів, для синхронізації або керування послідовністю виконання процесів і т.п.. Для цих цілей служать спеціальні системні програми, вбудовані в оболонку і підтримуючі наступні механізми взаємодії процесів :

- програми-сокети - для обміну даними;
- програмні канали для обміну даними між утилітами;
- асинхронні сигнали – ключові слова, передані від одного процесу до іншого, наприклад, для завершення виконання процесу або для одержання інформації про здійснення деяких подій у системі (сигнал від таймера, помилка при передачі даних на пристрій, спроба виконання неіснуючої команди й ін.);
- семафори – синхронізація виконання процесів за допомогою установки значень біт-"прапорів" спеціальних системних перемінних;
- спільно використовувана загальна область фізичної пам'яті.

Перераховані механізми використовує як сама ОС (write,mail і ін.), так і прикладні програми за допомогою команд керування процесами або системних викликів.

Розглянемо створення і роботу процесів на прикладі процедури. Тому що sh-процедура - процес, що виконується файл те, його виконання забезпечується в ОС звичайними механізмами побудови і керування процесами. Для кожної sh-процедури будується свій процес зі своїм дескриптором - породжений процес.

У дескрипторі процесу зберігається інформація, необхідна ядру для керування процесом. Дескриптори зберігаються в адресному просторі ядра ОС у вигляді пріоритетного списку, впорядкованого відповідно до їх номерів – ідентифікаторами процесів (PID).

Приклад 1

```
$ vi proc
date
ls
$sh proc
< дата>
f1
f2
$
```

Розглянемо як реалізуються процеси процедури прикладу.

У відповідь на запрошення -$ вводиться ім'я процедури proc і

створюється породжений процес shell. Він уводить дані, необхідні для свого виконання з зазначеного файлу proc - командні рядки. Кожна команда sh-процедури виконується породженим для неї процесом (як звичайно для команди, введеної з клавіатури - процеси утиліт). Як тільки всі команди виявляться виконаними, sh-процес завершується і керування повертається батьківському процесові. Процедура виконана.

Планувальник обробляє процеси двох видів – звичайні і більш пріоритетні процеси реального часу. Місце в черзі визначається його пріоритетом.

Процеси виконуються в одному з двох режимів - користувальному і у режимі ядра. У режимі ядра виконуються процеси системних викликів. У цьому випадку вони мають доступ до процедур і структур дані ядра. Така організація дозволяє ядру захистити свої процедури і структури системних даних від перекручувань з боку проблемних задач.

Дескриптори процесів створюються і знищуються динамічно при виконанні процесів. Тому склад і розмір списку постійно змінюється. Тому що всі процеси зв'язані між собою родинними відносинами, то спискова організація процесів використовується для процесів одного рангу (наприклад – для всіх породжених одним батьківським або для всіх батьківських і т.п.). У такий спосіб уся система дескрипторів являє собою деяку деревоподібну структуру з корінним загальносистемним процесом (swapper – ідентифікатор, що має, PID=0), що компілюється до складу ядра і використовується для побудови всієї структури дескрипторів і її обробки. Відкривається черга найбільш пріоритетним процесом ініціатора ОС (init – PID=1), що будується першим при ініціалізації ОС і знищується при завершенні роботи ОС.

Процесор виділяється кожному процесові по черзі на обмежені кванти часу. Це принципово відрізняє багатозадачний режим обробки завдань у UNIX від однозадачного режиму (DOS), де процесор виділяється процесам строго послідовно на увесь час виконання процесу. У UNIX перерахування пріоритетів відбувається постійно по перериваннях визначеного типу з частотою - біля одного разу в сек.

Частина інформації дескриптора може бути виведена в лістинг характеристик процесу за допомогою спеціальної команди ps (див. нижче). Деякі заголовки полів лістингу:

- номер термінала, якому належить процес (TTY)
- пріоритет (PRI)
- використаний час процесора (TIME)
- ідентифікатор процесу (PID)
- ім'я програми процесу або команди, виконуваної в тілі процедури на момент запиту лістингу (CMD)
- ідентифікатор батьківського процесу (PPID)
- адреса процесу (ADDR)
- величина зміни значення пріоритету (NI) і ін..

Повну інформацію про процес користувач або адміністратор може

одержати за допомогою цієї команди в наступному форматі:

$ps [-k] [<вхідне_ім'я_користувача>] - вивід лістингу характеристик процесу.

Деякі значення ключів:

-a[l] показати процеси даного термінала або

-af повний (достатній_ формат повідомлення

-afl довгий формат

-u показати всі активні процеси й ін..

Приклад 2

```
$ps -flu lev
PID   PPID   TTY   PRI   TIME        CMD
927    1      tty5         0:04          sh
1001  927     tty5         0:02          ps
..
```

Для виводу лістингу процесів, що належать користувачеві можна скористатися конструкцією:

$ps -f|grep <ім'я_користувача>.

2.2. Створення фонових процесів

При звичайному запуску з термінала деякої програми на виконання (системної утиліти або прикладної задачі) інтерпретатором shell створюється привілейований процес, що увесь час зв'язаний зі своїм терміналом. Запуск наступного процесу може бути виконаний користувачем тільки після завершення поточного, тобто з появою запрошення '$' від інтерпретатора.

З метою використання можливості рівнобіжного виконання програм в ОС UNIX окремі задачі або завдання пакетного режиму можуть бути запущені одночасно з завданнями діалогового режиму .Для запуску фонового (рівнобіжного з іншими нащадками) процесу в командний рядок необхідно і досить останнім символом додати знак & (амперсанд):

```
$ cc prog.c &
2388
$
```

Shell деколи виводить номер цього процесу (PID) і дозволяє введення наступної команди.

Фонові процеси мають деякі недоліки:

- не допускають уведення з клавіатури;

- забезпечують висновок на екран, але при цьому порушують цілісність висновку діалогового процесу.

Загальноприйнятий прийом виключення впливу фонового виводу на інтерактивну роботу:

– ком_рядок>ім'я_файлу.out &

– ком_рядок планує завдання для фонового режиму

– перенаправляє висновок замість екрана в зазначений файл головного каталогу користувача.

Приклад 3

```
$grep aaa* > grep.out &
194
$ps
PID          TTY          TIME          CMD
194          tty5          0:02          grep
 200          tty5          0:01          ps
```

Особливості роботи з фоновим режимом:

– виконувана у фоновим режимі програма (команда), що вимагає стандартного введення, повинна читати його з файлу з використанням перенаправленого виводу;

– програма, виконувана у фоновому режимі не може бути перервана <Ctrl*C>, тому що вона від'єднується від клавіатури і може бути припинена тільки за допомогою команди kill або виходом із системи;

– вихід із системи exit треба виконувати два рази: для завершення фонового процесу і завершення основного процесу shell.

У випадках, коли фоновий процес усе-таки вимагає введення даних із клавіатури, то його треба тимчасово перевести в оперативний режим, ввести дані, і повернути знову у фоновий за допомогою наступних команд:

fg %N - перевід фонового процесу в оперативний;

bg %N - перевід оперативного у фоновий режим.

Тут N – порядковий номер фонового завдання, що у загальному випадку може містити кілька активних процесів і усі вони переводяться у відповідний режим.

Номер завдання “N” виводиться: при запуску фонової програми командою jobs без призупинення або з призупиненням фонового процесу.

Призупинити виконання процесу з виходом у shell (наприклад для аналізу стану і результатів роботи процедури) можна за допомогою переривання -

<ctrl*Z>

$

с наступним запуском припиненого процесу.

Приклад 4

```
$inf>f.out&                  - запуск процедури inf у фоновому режимі
[1] 1754                     - номер завдання й ідентифікатор процесу
$jobs
 %1                           - номер завдання
<ctrl*Z>
[1]+stopped inf>f.out
$bg %1                       - запуск на продовження у фоновому режимі
[1]    inf>f.out
```

Виконання фонових завдань припиняється з виходом користувача із системи. АЛЕ! Якщо фонова програма повинна бути продовжена і після припинення поточного сеансу роботи, то необхідно використовувати команду:

$nohup ім'я_фонової_програми & команда запускає і захищає фонову програму від переривань, створюваних при виході користувача із системи, а також 2) перенаправляє фоновий протокол у системний файл nohup.out.

Разом з тим, висновок протоколу роботи фонової програми можна перенаправляти в спеціальний файл, що переглянути пізніше без порушення протоколу роботи з оперативною задачею:

$nohup ім'я_тло_програми>ім'я_файлу&.

Розглянемо два випадки застосування команди nohup.

Перший приклад, де nohup перенаправляє протокол в вказаний файл :

```
$nohup cat * > outfile &
[1] 972
$exit            - завершується робота оперативного процесу

Повторний вхід у систему:
login:
password:
$ps  -af|grep cat          - конвеєр, виділяється рядок із шаблоном cat
UID       PID    PPID          CMD
lev       972      1        cat *>outfile &
$
```

У прикладі "процес йде", nohup стає самостійним процесом і його PPID міняється - батьківським процесом були sh (див. приклади вище), а став init с PID=1, тобто пріоритет фонового процесу "підсилюється" загальносистемним процесом init і його виконання буде продовжуватися.

Примітка: у поле CMD може не вказуватися ім'я процедури, а виводитися ім'я поточної утиліти цієї процедури, ідентифікувати процедуру краще по PID.

Якщо перенаправлення протоколу не використовувати, то вивід автоматично здійснюється в системний файл nohup.out, створюваний системою в головному каталозі користувача HOME.

2.3. Керування пріоритетами

Максимальний пріоритет процесів кожного користувача групи встановлює адміністратор. Якщо при виконанні завдання утворяться кілька породжених процесів, то усі вони мають однаковий пріоритет рівний батьківському. У цьому випадку всі процеси одержують ресурси рівними частками (простий режим поділу часу). При необхідності виділення найбільш важливих батьківських процесів породженим другорядним можна понизити пріоритет за допомогою команди:

nice [-k] ім'я_програми виконання програми, яка зазначена у рядку, зі зниженим пріоритетом.

-k - к-т зниження пріоритету (k = 1;...10; за замовчуванням k = 10).

Приклад 5

$ prog1 &
$ ps -al
..... PID ..... PRI NI.....CMD..
..... 20 sh
..... . 20 prog1.....
батьківський і породжений процеси мають пріоритет=20

Приклад 6

$ nice -5 prog2 &
[2] 3752
$ps -flu lev
..... PID ..... PRI .....NI.. CMD
..... . 20 ..... sh
..... 3752 ..... 25 .....5 prog2.
..... 20.. .....

Пріоритет процесу задачі prog2 знижений на 5 одиниць, тим самим інші процеси цієї групи більше можливостей у використанні ресурсів.
Чим більше число, тим нижче пріоритет.

2.4. Завершення процесів

Завершення процесів – одна з функцій керування процесами. Припинити виконання будь-якого процесу можна за допомогою команди:

kill [-опції] PID1 [PID2.....] - передає сигнал процесові PID

Сигнал - ключове слово, при одержанні якого процес виконує деякі дії. Сигнали з використанням команди kill можуть передаватися іншими прикладними процесами або системними програмами (наприклад з появою деяких подій, як те збій у каналі, сигналу з таймера, завершення фонового процесу й ін.). Існує двадцять п'ять видів сигналів, призначених для виконання різних дій процесами при настанні визначених подій у системі. З одержанням більшості сигналів процес завершується самостійно. Той хто посилає сигнал повинний бути власником процесу або адміністратором. Для безумовного і негайного завершення зазначеного процесу kill повинен послати сигнал з ім'ям TERM (за замовчуванням). Інші сигнали передаються за допомогою опції -S.

Значення опцій:

-S<ім'я_сигналу> або
-№ - системний номер сигналу
-l - вивід на екран довідника імен сигналів

Сигнал визначає подальшу дію процесу. Це ще один спосіб керування процесами (див. вище).

Приклади опцій:

-S KILL - негайне завершення процесу по сигналі KILL
-9 - те ж, але вже по номеру сигналу KILL

Приклад 7

```
$kill   [-9]    3752
3752: killed
$
```

Приклад завершення фонового процесу:

```
$kill %1                    - указується номер завдання, що завершується
[1]    +Terminated  inf
$
```

Якщо процес зазначений ідентифікатором "0", то команда kill знищує всі процеси, зв'язані з поточної shell-ом (завершується головний процес swapper (що має PID=0)) і усі зв’язані з ним породжені процеси – init, shell, прикладні задачі).

2.5. Перехоплювання сигналів

У процедурах керування процесами особливе місце займає команда:

trap ' список команд або ім'я sh-процедури'   сигнал1   сигнал2

Команда перехоплює зазначені сигнали і послідовно виконує всі команди списку перш ніж сигнал, що надійшов, буде переданий процесові. Наприклад, команда може бути використана для синхронізації завдань, якщо треба виконати які те дії з файлами, а потім уже завершити процес.

Приклад 8. Циклічна процедура виконується до надходження одного з зазначених сигналів.

```
$cat > trapfil
trap 'echo  натиснути Ctrl*C'  INT  QUIT TERM
while  true
do
      echo цикл
done
$sh  trapfil
цикл
цикл
……....
натиснути Ctrl*C                - надійшов один із зазначених трьох сигналів
<ctrl*C>
$
```

– циклічна процедура буде довершена при надходженні хоча б одного з зазначених сигналів.

Застосування команди trap може бути найрізноманітнішим. Але глибоке її вивчення вимагає попереднього знання всього списку сигналів і їхнього призначення, а також найбільше що часто зустрічаються типових застосувань trap.

Завдання 2

Робота має на меті - закріпити представлення про можливості командної мови Unix/Linux по керуванню процесами, яким виділяються всі необхідні ресурси обчислювальної системи.

Вивчаються команди:

ps - запит інформації про процеси поточного термінала;
& - запуск фонового процесу;
fg, bg - переводить процес в активний або фоновий режим;
jobs - запит лістингу списку завдань;
nohup - захист фонових процесів від переривання виконання при виході із сеансу
роботи із системою;
nice - зниження пріоритету процесу;
kill - припинення виконання процесу.

Методика виконання завдання 2

1. Вивести на екран лістинг характеристик (у довгому і короткому форматах) процесів, ініціалізованих з вашого термінала. Проаналізувати і пояснити зміст кожного поля повідомлення.
2. Розробити і запустити найпростішу процедуру у фоновому режимі з нескінченним циклом виконання, що передбачає, наприклад, перенапрямок виводу яких не будь повідомлень у файл або у фіктивний файл, і команду, що використовує, sleep для скорочення частоти циклів процедури.
3. Виконати п. 1. Пояснити зміни в лістингу характеристик процесів. Пояснити зміст PID і PPID.
4. Знизити значення пріоритету процедури. На що і як вплине ця операція при керуванням обчислювальним процесом системи?
5. Проаналізуйте лістинг процесів. Який процес є батьківським для процедури.
6. Вийдіть із системи і ввійдіть заново. Проаналізуйте лістинг процесів. Поясните зміни в системі.
7. Запустіть процедуру у фоновому режимі, але необхідно передбачити її захист від переривання при виході із системи..
8. Виконаєте п.6. Поясніть зміни PPID процедури.
9. Завершіть виконання процесу процедури.
10. Запустіть процедуру в оперативному режимі з перенапрямком виводу у відповідний файл.
11. Переведіть завдання з процедурою у фоновий режим і проаналізуйте повідомлення на екрані.
12. Переведіть завдання з процедурою в оперативний режим і проаналізуйте повідомлення на екрані.
13. Завершите виконання процедури і проаналізуйте повідомлення на екрані.

14. Проаналізуйте з використанням команди history зміст лабораторної роботи, продумайте відповіді на контрольні запитання і здайте виконану роботу викладачеві.

3. Контрольні запитання

1. Поясніть поняття процесу і ресурсу. Яке їхнє значення в організації обчислювального процесу в ОС UNIX?
2. Якими способами можна організувати виконання програм у фоновому режимі?
3. Які особливості виконання програм у фоновому режимі? Як уникнути виводу фонових повідомлень на екран і переривання виконання фонових програм при припиненні сеансу роботи із системою?
4. Як користувач може вплинути на розподіл ресурсів між активними процесами?
5. Як можна перервати виконання активних процесів? Яка інформація для цього необхідна і відкіля вона витягається?

ЛАБОРАТОРНА РОБОТА 8

Тема: Управління пам'яттю.

Мета роботи: одержання практичних навичок керування пам'яттю і самостійною роботою з документацією команд.

1. Основні теоретичні відомості

Команди POSIX для роботи з пам'яттю (повинні бути у всіх операційних системах)

ps - виводить інформацію про процеси і пам'ять

Команди для роботи з LINUX

free - виводить інформацію про використання оперативної пам’яті

top - виводить динамічну інформацію про процеси і пам’ять

ps - виводить інформацію про процеси і пам'ять

Команди для роботи з Windows

Велику частину інформації про пам’ять можна одержати через **диспетчер задач**.

Tasklist - виводить інформацію про працюючі процеси і пам'ять

2. Порядок роботи

Завдання 1

Запустіть Linux.

1. Складіть довідник для вище наведених команд (українською мовою), розписавши які параметри для чого потрібні.
2. Попрацюйте з цими командами.
3. Що потрібно вміти:
   - виводити інформацію про використання оперативної пам'яті, періодично і з різними одиницями виміру;
   - виводити інформацію про пам'яті процесів, і розуміти, що означає той або інший стовпець.

Завдання 2

Запустіть Windows

1. Складіть довідник для вище наведених команд (українською мовою), розписавши які параметри для чого потрібні.
2. Розберіться як працювати з диспетчером задач, і що означає інформація в стовпцях (усі що відноситься до пам'яті).
3. Попрацюйте з цими командами.
4. Що потрібно вміти:
   - показувати робочий набір для кожного процесу й у цілому для системи;

- зміна робочого набору з моменту останнього відновлення для кожного процесу;
- пам’ять, що не вивантажується, для кожного процесу й у цілому для системи;
- пам’ять, що вивантажується, для кожного процесу й у цілому для системи;
- число звертань до диска для завантаження сторінок, не знайдених в для кожного процесу;
- зміна числа звертань до дискові для завантаження сторінок, не знайдених в ОЗУ, з моменту останнього відновлення;
- показувати максимальний робочий набір для кожного процесу
- розмір адресного простору, переданого процесові .

До здачі лабораторної здати: довідник команд, що надають інформацію про пам’ять.

ЗМІСТ