

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
КРЕМЕНЧУЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ МИХАЙЛА ОСТРОГРАДСЬКОГО



МЕТОДИЧНІ ВКАЗІВКИ  
ЩОДО ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ  
З НАВЧАЛЬНОЇ ДИСЦИПЛІНИ  
**«ОБЧИСЛЮВАЛЬНА ТЕХНІКА»**  
ДЛЯ СТУДЕНТІВ ДЕННОЇ ФОРМИ НАВЧАННЯ  
ЗА НАПРЯМОМ 123 – «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ»  
ЧАСТИНА II

КРЕМЕНЧУК 2018

Методичні вказівки щодо виконання лабораторних робіт з навчальної дисципліни «Обчислювальна техніка» для студентів денної форми навчання за напрямом 123 – «Комп'ютерна інженерія». Частина II

Укладачі: старш. викл. Ю. В. Зілінський,  
к. т. н., доц. О. Г. Славко

Рецензент к. т. н., доц. В. М. Сидоренко

Кафедра комп'ютерних та інформаційних систем

Затверджено методичною радою КрНУ імені Михайла Остроградського  
Протокол № \_\_\_\_\_ від \_\_\_\_\_ 2018 р.

Голова методичної ради \_\_\_\_\_ проф. В. В. Костін

## ЗМІСТ

Вступ.....	4
1 Перелік лабораторних робіт.....	6
Лабораторна робота № 1 Файловий менеджер FAR Manager.....	6
Лабораторна робота № 2 Сценарії командної оболонки операційної системи Windows NT.....	19
Лабораторна робота № 3 Командний інтерпретатор Linux.....	32
Лабораторна робота № 4 Файловий менеджер GNU Midnight Commander і сценарії оболонок Linux.....	38
2 Критерії оцінювання якості виконання лабораторних робіт студентами.....	44
Список літератури.....	46

## ВСТУП

Навчальний курс є базовим нормативним навчальним курсом, що викладається в першому навчальному семестрі для студентів денної форми навчання за напрямом 123 – «Комп'ютерна інженерія». Перша частина даних методичних вказівок охоплює перші сім робіт, передбачених для виконання робочою навчальною програмою з навчальної дисципліни «Обчислювальна техніка».

Предметом вивчення навчальної дисципліни є інформаційні, арифметичні, логічні й алгоритмічні основи комп'ютерів, а також принципи організації комп'ютерів, архітектура персональних комп'ютерів, організація пам'яті комп'ютера, файлових систем та інше, що надає повне уявлення про процес функціонування сучасних комп'ютерів.

Дані методичні вказівки, зокрема частина I, дозволяють студентам ознайомитися з можливостями, які надаються операційними системами (ОС) Windows за допомогою інтерфейсу командного рядка. Ця частина сервісу ОС є майже незнайомою для пересічних користувачів.

Хоча сімейство операційних систем Microsoft Windows набуло популярності, насамперед завдяки розвиненому графічному інтерфейсу, усі версії Microsoft Windows підтримують і засоби командного інтерфейсу, як це було ще за часів ОС MS DOS. Розвиток зручних для користувача графічних оболонок та інтерфейсів операційних систем Windows 3.x, а згодом і Windows 9x не призупинив розвиток командного інтерфейсу. Кожна наступна версія ОС від Microsoft розширювала можливості консольного вікна командного інтерпретатора, додаючи до них нові засоби.

Набір команд Windows NT значно розширився, порівняно з Windows 9x. Окрім того, що Windows NT набула специфічних тільки для неї команд, вона також позбулася незначної кількості команд, які не відповідають принципам її побудови. Власний набір команд Windows NT уже недоступний з командного рядка MS DOS чи Windows 9x і може виконуватися лише в середовищі

Windows NT, однак командний рядок MS DOS і досі може стати в нагоді. Сучасні комп'ютери на базі новітніх мікропроцесорів виконують процес завантаження так, як і попередні моделі, що дозволяє завжди завантажити DOS-подібну операційну систему із зовнішнього накопичувача для розв'язання завдань технічної діагностики та обслуговування апаратної частини.

**Мета і завдання навчальної дисципліни:** набуття студентами загальних теоретичних і практичних знань у галузі обчислювальної техніки, а також отримання знань специфіки власного набору команд Windows NT, виконання діагностики апаратної частини комп'ютера, засвоєння засобів адміністрування операційних систем за допомогою інтерфейсу командного рядка та ін.

**Місце навчальної дисципліни у навчальному процесі:** курс «Обчислювальна техніка» тісно пов'язаний з такими дисциплінами, як «Комп'ютерні системи», «Комп'ютерні мережі», «Проектування та адміністрування комп'ютерних мереж», «Програмування», «Прикладна теорія цифрових автоматів», «Системне програмування», «Архітектура комп'ютерів».

**У результаті вивчення дисципліни студент повинен**

**знати:**

- теоретичні та практичні знання у галузі обчислювальної техніки;
- загальні можливості, які надаються операційними системами Windows за допомогою інтерфейсу командного рядка;
- засоби командного інтерпретатора Windows NT;
- специфіку організації зберігання даних та роботи з ними;

**уміти:**

- виконувати технічну діагностику та обслуговування апаратної частини комп'ютера;
- виконувати аналіз і тестування мережних процесів засобами командної оболонки ОС Windows NT;
- використовувати внутрішні та зовнішні команди командного інтерпретатора Windows для роботи з даними та виконання діагностування стану комп'ютера.

# 1 ПЕРЕЛІК ЛАБОРАТОРНИХ РОБІТ

## Лабораторна робота № 1

### Тема. Файловий менеджер FAR Manager

**Мета:** набуття практичних навичок роботи з файловим менеджером FAR Manager в операційних системах Windows NT і подібними до нього оболонками операційних систем DOS/Windows/Linux.

### Короткі теоретичні відомості

Файлові менеджери належать до тих програм, які обов'язково використовуються в повсякденній роботі. Головним призначенням файлового менеджера (англ. file manager) за часів, коли основним інтерфейсом операційних систем був командний рядок, було надання зручного інтерфейсу користувача для роботи з файловою системою для виконання найбільш частих операцій над файлами – переміщення, перейменування, копіювання, видалення, зміна атрибутів і властивостей, пошук файлів і призначення прав, створення, відкриття, програвання, перегляд, редагування.

Сучасні операційні системи, зазвичай, мають власні файлові менеджери, однак вони не вирізняються широкими функціональними можливостями, що не зменшило й сьогодні потреби у файлових менеджерах сторонніх розробників. Серед великої кількості сучасних файлових менеджерів для ОС Windows лідерство можна надати двом програмам – FAR Manager і Total Commander.

Розглядаючи файловий менеджер як допоміжний засіб адміністрування системи, для вивчення в лабораторному практикумі було вибрано FAR Manager. Серед причин такого вибору можна зазначити декілька.

Автор програми Євген Рошал свого часу безкоштовно передав вихідні тексти розробленого ним файлового та архівного менеджера групі розробників з так званої FAR Group, і сьогодні Far Manager є вільно поширюваним програмним забезпеченням (<http://farmanager.com/>), яке з 26 жовтня 2007 року випускається FAR Group з модифікованою ліцензією BSD.

Far Manager має гнучку модульну структуру, і його функціональність

може бути істотно розширена за допомогою зовнішніх DLL-модулів – плагінів, переважна більшість яких також поширюється вільно. Деякі з найбільш цікавих і важливих плагінів безпосередньо долучено до складу стандартної поставки Far Manager, а будь-які інші плагіни користувач може легко встановити, самостійно вибираючи їх перелік відповідно до своїх потреб (<http://forum.farmanager.com/>).

Far Manager розроблено у вигляді консольного додатка Windows, що передбачає менші вимоги до системних ресурсів. Far Manager підтримує різні кодування текстів і має багатомовний інтерфейс та систему допомоги.

З 2007 року Far Manager розробляється як Unicode-додаток, і він позбувся недоліків, пов'язаних з використанням на консолі української літери «і». Far Manager підтримує всі можливості файлової системи NTFS (довгі Unicode-імена файлів, NTFS-атрибути файлів, альтернативні потоки даних, жорсткі зв'язки та символічні посилання). FAR Manager працює з будь-якою версією Windows NT і паралельно розробляється FAR Group у двох варіантах – 32- та 64-розрядному.

FAR Manager підтримує двовіконну концепцію, стандартну (за замовчанням) кольорову схему та систему команд керування з клавіатури повністю сумісну з відомим файловим менеджером Norton Commander (або подібними до нього Volkov Commander і DOS Navigator) для ОС DOS і найпопулярнішим файловим менеджером Midnight Commander для ОС Linux, що дозволяє легко перенести практичні навички, набуті під час роботи з FAR Manager, для освоєння інших файлових менеджерів у середовищі інших операційних систем.

У цій лабораторній роботі складовою частиною звіту є протокол сеансу роботи з файловим менеджером. Деякі пункти розділу «Порядок виконання роботи» містять завдання, що можуть бути виконані декількома способами. Якщо в пункті завдання безпосередньо не зазначено використовувати для його виконання засоби та/або методи – їх самостійно вибирає студент.

Загалом будь-який один запис протоколу повинен містити номер пункту

порядку виконання роботи і послідовність дій (алгоритм), використану для виконання завдання цього пункту.

У деяких випадках послідовність дій і використовувані засоби жорстко визначаються безпосередньо в пункті порядку виконання роботи. У такому випадку студент виконує роль експериментатора, який повинен відтворити експеримент згідно із запропонованою схемою, провести спостереження і зробити висновки щодо його результатів. У таких пунктах порядку виконання роботи обов'язково зазначається, що повинно бути відображено у звіті.

Приклад 1. Фрагмент протоколу сеансу роботи з файловим менеджером Far під час виконання пунктів 11, 12 і 48 поточної лабораторної роботи (див. порядок виконання роботи).

11. Клавiша TAB призначена для перемiщення екранного курсора мiж панелями Far Manager. Натискання клавiшi TAB перемiщує екранний курсор на протилежну панель (з правої на лiву або навпаки), що дозволяє змiнити активну панель. З урахуванням того, що рiзні панелі можуть бути настроєні на вiдображення рiзних дискiв і каталогiв, натискання клавiшi TAB еквiвалентне у цьому випадку командам змiни поточного диска і каталогу в середовищі командного iнтерпретатора Windows.

12. Передостанній рядок вiкна Far Manager дозволяє виконати введення тексту з клавiатури у позицію, на яку вказує текстовий курсор. Пiсля натискання клавiшi Enter уведений текст iнтерпретується як команда. Far Manager має власний командний iнтерпретатор і набiр команд. Якщо Far Manager не розпiзнає власну команду, він вважає, що це команда операцiйної системи і передає її на виконання командному iнтерпретаторовi Windows. З введенням команди DIR вона виконується командним iнтерпретатором Windows вiдносно того каталогу, який на цей час був поточним на активній панелi Far Manager. Вiдгук командного iнтерпретатора Windows на команду вiдображається на консолi Far Manager. Для перегляду консолi Far Manager потрiбно тимчасово прибрати вiдображення файлових панелей, що дозволяє зробити натискання комбiнацiї клавiш Ctrl-O. Повторне натискання цiєї



комбінації клавіш відновлює відображення файлових панелей Far Manager.

48. На лівій панелі робимо поточним каталог WINCMD\HELP поточного каталогу, установивши екранний курсор спочатку на каталог WINCMD і натиснувши Enter, а потім установивши екранний курсор на каталог HELP і натиснувши Enter. Натиснувши комбінацію клавіш Alt-F1, вибираємо у вікні діалогу диск C:. За допомогою клавіші TAB робимо активною праву панель і (за необхідності) переміщуємося на правій панелі у кореневий каталог диска C: за допомогою комбінації клавіш Ctrl-\. Установивши екранний курсор на каталог Windows і натиснувши Enter, робимо поточним каталог Windows. Установивши екранний курсор на каталог Help і натиснувши Enter, робимо поточним каталог Help. Позначаємо на правій панелі файли, призначені для копіювання, натиснувши клавішу G+, зазначивши у вікні діалогу маску \*.cnt і натиснувши Enter. Виконуємо копіювання, натиснувши спочатку клавішу F5, а потім клавішу Enter.

Протокол дозволяє викладачеві перевірити правильність виконання студентом відповідних пунктів порядку виконання роботи й повинен бути логічно пов'язаний і відтворюваний, тобто послідовне виконання зазначених у протоколі дій на «живій» системі повинно призводити до виконання зазначених у пунктах порядку виконання роботи завдань.

### **Хід роботи**

1. Увімкнути комп'ютер, завантажити операційну систему Windows NT, увійти до свого облікового запису і відкрити сеанс роботи командного інтерпретатора ОС.

2. Видалити підкаталог "%USERPROFILE%\Мои документы\Моя освіта", створений під час виконання попередніх лабораторних робіт.

3. Використовуючи команду DIR з відповідними опціями та комбінуючи її з командами-фільтрами для організації конвеєрів, знайти на магнітному диску каталог установлення файлового та архівного менеджера Far Manager за іменем основного програмного модуля Far.exe (зазвичай це каталог %ProgramFiles%\Far).

4. Зробивши каталог установлення Far Manager поточним, виконати настроювання його параметрів експортом у реєстр файлів FarColors.reg і FarHKCU.reg за допомогою редактора реєстра. Експорт у реєстр reg-файлу з іменем filename.reg виконується з командного рядка командою REGEDIT /S filename.reg.

5. Завершити сеанс роботи командного інтерпретатора за допомогою команди EXIT і виконати запуск Far Manager (програмний файл Far.exe) будь-якими стандартними засобами операційної системи. Наприклад, натиснути комбінацію клавіш Win-R, увести %ProgramFiles%\Far\far.exe і натиснути Enter. Для збереження виконуваних у наступному пункті настроювань параметрів консольного вікна Far Manager для його запуску слід вибрати той метод, який буде використовуватися для цього й надалі.

6. Виконати настроювання параметрів консольного вікна Far Manager:

а) натисніть на лівому верхньому куті вікна й виберіть команду «Свойства»;

б) виберіть вкладку «Расположение» і в областях «Размер буфера экрана» та «Размер окна» введіть або виберіть значення 80 у полі «Высота» і 25 у полі «Ширина»;

в) зніміть прапорець «Автоматический выбор» в області «Положение окна», а потім установіть значення -4 в полях «Левый край» і «Верхний край»;

г) виберіть вкладку «Шрифт» і виберіть Lucida Console в області «Шрифт», а в області «Размер» збільшіть значення так, щоб консольне вікно займало всю ширину екрана;

д) натисніть кнопку «ОК» і у діалоговому вікні «Изменение свойств» виберіть варіант «Сохранить свойства для других окон с тем же именем».

7. Ознайомитися з правилами використання довідкової системи Far Manager. Довідкова система викликається за допомогою клавіші F1. Довідка є контекстно-чутливою, тобто завжди надається та інформація, яка безпосередньо стосується поточного стану вікна програми. Правила навігації по сторінках довідкової системи наведені в другому пункті основного вікна

довідки.

8. Розпочати ведення протоколу сеансу роботи з Far Manager з наступного пункту порядку виконання роботи.

9. Натиснувши клавішу F9, активізувати меню Far Manager і виконати настроювання параметрів за допомогою однойменного пункту меню. Настроювання параметрів можна виконати на свій розсуд, але обов'язково встановити:

- а) видалення файлів у Кошик;
- б) відображення прихованих і системних файлів;
- в) автоматичну зміну папки з переміщенням по дереву диска;
- г) виведення всіх попереджувальних повідомлень.

10. Натиснути декілька раз клавішу TAB і занести до звіту її призначення.

11. Увести з клавіатури слово «DIR», а потім натиснути клавішу Enter. Натиснути декілька разів комбінацію клавіш Ctrl-O і клавішу ESC. Зазначити у звіті призначення передостаннього рядка вікна Far Manager і призначення комбінації клавіш Ctrl-O.

12. Натиснути по чергово комбінації клавіш Alt-F1 та Alt-F2. У меню, що з'являється в результаті натискання комбінацій цих клавіш, вибрати «C:» у лівій панелі та «D:» (по можливості) у правій панелі, натискаючи клавішу Enter. Зазначити у звіті призначення комбінацій клавіш Alt-F1 та Alt-F2.

13. Змінюючи по чергово за допомогою клавіші TAB положення екранного курсора Far Manager з правої на ліву панель, натиснути комбінацію клавіш Ctrl-L і занести до звіту її призначення.

14. Установивши за допомогою клавіш «↓» та «↑» екранний курсор Far Manager на каталог %ProgramFiles%, на лівій або правій панелі екрана натиснути по чергово:

- а) три рази клавішу Enter;
- б) два рази клавішу PageDown, два рази клавішу PageUp;
- в) два рази клавішу «→», два рази клавішу «←»;
- г) клавішу «→», а потім клавішу PageUp.

15. Зазначити у звіті призначення клавіш «↓», «↑», «→», «←», PageDown, PageUp та Enter під час роботи з каталогами на файловій панелі Far Manager згідно з результатами виконання попереднього пункту.

16. Установивши екранний курсор Far Manager на каталог %ProgramFiles% на лівій або правій панелі, натиснути по чергово клавіші F3, F4 та комбінацію клавіш Ctrl-Q і занести до звіту їх призначення.

17. Перейти на лівій панелі у каталог «%USERPROFILE%\Мои документы», чотири рази натиснути клавішу Backspace і занести до звіту її призначення.

18. Повернутися на лівій панелі у каталог «%USERPROFILE%\Мои документы», натиснути комбінацію клавіш Ctrl-\ і занести до звіту її призначення.

19. Знову повернутися на лівій панелі у каталог «%USERPROFILE%\Мои документы» і натиснути комбінацію клавіш Ctrl-Shift-1. Після цього натиснути комбінацію клавіш Ctrl-\, а потім – комбінацію клавіш RCtrl-1 (правий Ctrl і 1). Занести до звіту призначення комбінацій клавіш Ctrl-Shift-цифра і RCtrl-цифра.

20. Створити (F7) в каталозі «Мои документы» підкаталог «Моя освіта», а в ньому – наступну структуру каталогів (рис. 1).

21. Установити атрибут (Ctrl-A) «прихований» каталогам «Відпочинок і дозвілля» і «2-й семестр». Занести до звіту призначення комбінацій клавіш Ctrl-A.

22. Зробивши в лівій панелі поточний каталог «Моя освіта», натиснути комбінацію клавіш Ctrl-T і перейти на правій панелі на дерево підкаталогів каталогу «Моя освіта». Занести до звіту призначення комбінацій клавіш Ctrl-T.

23. Переміщуючись по дереву каталогів на правій панелі, зробити поточним каталог «%USERPROFILE%\Мои документы \ Моя освіта \ Університет \ Навчання \ 1-й курс \ 1-й семестр \ Обчислювальна техніка \ Лабораторні роботи». У подальшому під час виконання даної лабораторної роботи цей каталог вважають за поточний.



Рис. 1 – Структура каталогів

24. Повернути екранний курсор на ліву панель і зняти за допомогою комбінації клавіш Ctrl-T відображення дерева каталогів у правій панелі.

25. Перейти на лівій панелі у кореневий каталог поточного диска.

26. Натиснувши комбінацію клавіш Alt-F10 і дочекавшись вікна дерева каталогів, розпочати введення у рядок пошуку імені «Лабораторні роботи», доки екранний курсор не встановиться на каталозі «%USERPROFILE% \ Мои документы \ Моя освіта \ Університет \ Навчання \ 1-й курс \ 1-й семестр \ Обчислювальна техніка \ Лабораторні роботи». Натиснувши клавішу Enter, перейти у лівій панелі у цей каталог.

27. Натиснувши комбінацію клавіш Shift-F4, створити в поточному каталозі текстовий файл у кодуванні Windows-1251 з іменем Description з трьох рядків тексту. Перший рядок – «Лабораторні роботи з навчальної дисципліни «Обчислювальна техніка»», другий рядок – своє прізвище, ім'я та по батькові у родовому відмінку, третій рядок – «виконані у 1-му семестрі 1-го курсу». Після закінчення редагування зберегти файл, натиснувши клавішу F2.

28. Створити (Shift-F4) по чергово в поточному каталозі текстові файли fio-ansi.txt, fio-koi8.txt, fio-iso.txt, fio-utf.txt, записавши у них своє прізвище, ім'я

і по батькові у кодуваннях CP1251, KOI-8U, ISO 8859-5, Unicode відповідно. Вибір кодування можна зробити як у вікні діалогу створення файлу, так і безпосередньо у самому редакторі за допомогою Shift-F8.

29. Переглянути (F3) створені у попередньому пункті текстові файли fio-ansi.txt, fio-koi8.txt, fio-iso.txt, fio-utf.txt у шістнадцятковому (F4) вигляді й занести до звіту своє прізвище, ім'я і по батькові в символному вигляді та навести еквівалентні цьому рядку символів послідовності байтів у кодуваннях CP1251, KOI-8U, ISO 8859-5, Unicode.

30. Установити екранний курсор на перший зі створених txt-файлів і натиснути чотири рази клавішу Insert.

31. Натиснути клавішу G- («мінус» на додатковій цифровій клавіатурі з права), у вікні діалогу ввести шаблон \*.txt і натиснути Enter.

32. Повторити попередній пункт для клавіші G+, а потім натиснути 2 рази клавішу G\* і за результатами виконання трьох попередніх пунктів занести до звіту призначення клавіш Insert, G+, G- та G\*.

33. Зробити копії (F5) файлів fio-ansi.txt, fio-koi8.txt, fio-iso.txt, fio-utf.txt у файлах fio-ansi.bak, fio-koi8.bak, fio-iso.bak, fio-utf.bak у тому самому каталозі.

34. Створити жорсткий зв'язок (Alt-F6) для файлу fio-ansi.txt з іменем fio-ansi.hln у тому самому каталозі.

35. Виконати редагування (F4) файлу fio-ansi.txt, а потім переглянути вміст (F3) файлу fio-ansi.hln.

36. Виконати редагування файлу fio-ansi.hln, а потім переглянути вміст файлу fio-ansi.txt. Зазначити у звіті, як зміни одного з файлів позначаються на іншому.

37. Установити всім файлам з розширенням «txt» поточного каталогу атрибут (Ctrl-A) «тільки для читання», а файлу Descript.ion поточного каталогу – атрибут «прихований».

38. Створити (F7) у поточному каталозі три підкаталоги з іменами «BAK», «DOC», «WINCMD».

39. Скопіювати всі файли з розширенням «txt» поточного каталогу в

підкаталог DOC поточного каталогу.

40. Змінити розширення (F6) усім файлам підкаталогу DOC поточного каталогу з «txt» на «doc».

41. Перемістити (F6) усі файли з розширенням «bak» поточного каталогу у підкаталог BAK поточного каталогу.

42. Видалити (F8) усі файли з розширенням «txt» поточного каталогу.

43. Створити (F7) у підкаталозі WINCMD поточного каталогу три підкаталоги з іменами «HELP», «SYS», «TEMP».

44. Створити зв'язок (Alt-F6) для каталогу BAK у підкаталозі WINCMD.

45. Виконати редагування (F4) будь-якого файлу каталогу WINCMD\BAK і переглянути вміст цього самого файлу у вихідному каталозі BAK.

46. Створити у вихідному каталозі BAK або у зв'язаному з ним каталозі WINCMD\BAK будь-який файл. Переглянути вміст обох каталогів і зазначити у звіті, як зміни одного з каталогів позначаються на іншому каталозі.

47. Видалити (F8) у вихідному каталозі BAK створений під час виконання попереднього пункту файл і видалити зв'язок для каталогу BAK у підкаталозі WINCMD.

48. Скопіювати в підкаталог HELP каталогу WINCMD файли підкаталогу Help каталогу встановлення Windows з розширенням «cnt».

49. Скопіювати в підкаталог HELP каталогу WINCMD файли підкаталогу Help каталогу встановлення Windows, імена яких починаються з послідовності символів «new».

50. Скопіювати в підкаталог SYS каталогу WINCMD файли системного (System32) підкаталогу Windows з розширенням «exe» та іменами з двох символів.

51. Скопіювати в підкаталог SYS каталогу WINCMD файли системного підкаталогу Windows з розширенням «dll» та іменами, які починаються з послідовності символів «ms», а закінчуються на «32».

52. Скопіювати в підкаталог SYS каталогу WINCMD файли системного підкаталогу Windows з розширенням із двох символів, перший з яких «a», та з

іменами, що починаються із символу «m».

53. Перемістити всі файли з розширенням «dll» з підкаталогу «SYS» каталогу WINCMD у підкаталог TEMP каталогу WINCMD.

54. Перемістити підкаталог SYS каталогу WINCMD в підкаталог TEMP каталогу WINCMD.

55. Запакувати (Shift-F1) в архів 7z з ім'ям «help» усі файли підкаталогу HELP каталогу WINCMD. Архівний файл розмістити в тому самому каталозі, що і вихідні файли.

56. Видалити всі файли підкаталогу HELP каталогу WINCMD, окрім архівного.

57. Розпакувати з архіву з ім'ям «help» підкаталогу HELP каталогу WINCMD усі файли в підкаталог TEMP каталогу WINCMD.

58. Видалити з підкаталогів HELP та TEMP каталогу WINCMD усі файли та каталоги.

59. Створити копію каталогу DOC з ім'ям «TXT» у тому самому каталозі.

60. Порівняти (F9→Команды→Сравнение папок) вміст каталогів DOC і TXT.

61. Створити в будь-якому з каталогів DOC і TXT ще один текстовий файл і порівняти вміст каталогів DOC і TXT.

62. Додати в будь-який текстовий файл у будь-якому з каталогів DOC і TXT будь-який рядок символів і порівняти вміст каталогів DOC і TXT.

63. Змінити час модифікації (Ctrl-A) на більш пізній будь-якого текстового файлу у будь-якому з каталогів DOC і TXT та виконати розширене порівняння (F11→Расширенное сравнение) каталогів DOC і TXT.

64. Знайти (Alt-F7 і Фільтр) і вивести інформацію про:

а) усі можливі місця розташування файлів з іменем «dbghelp.dll» на системному диску;

б) кількість усіх файлів з розширенням «exe» в каталозі встановлення Windows;

в) загальну кількість усіх підкаталогів каталогу встановлення Windows;



г) усі файли з розширенням «exe», створені у системному каталозі Windows протягом трьох минулих місяців;

д) усі файли на диску С, які були модифіковані учора з 15 до 16 години;

е) усі програми каталогу %ProgramFiles%, які виконувалися сьогодні (до файлів яких виконувався доступ).

65. Знайти у каталозі «%USERPROFILE%\ Мои документи \ Моя освіта \ Університет \ Навчання \ 1-й курс \ 1-й семестр \ Обчислювальна техніка \ Лабораторні роботи» і всіх його підкаталогах усі файли, що містять ваше прізвище, ім'я та по батькові.

66. Виконати з командного рядка Far запуск додатка Блокнот (notepad.exe), а потім знищити процес (Ctrl-W) для notepad.exe.

67. Виконати запуск додатка Блокнот. Переглянути (Alt-F1/F2→Список процессов) властивості процесу з образом notepad.exe (курсор на відповідний рядок і F3/F4) та знищити (F8) цей процес. Увага! Це не стандартна можливість Far, яка реалізується плагіном.

68. Переглянути всі активні мережні з'єднання (Alt-F1/F2→Сетевые соединения). Увага! Це також не стандартна можливість Far, яка реалізується плагіном. Якщо розмір панелі Far виявляється недостатній для відображення всієї інформації, можна змінити співвідношення панелей за допомогою комбінації клавіш Ctrl-→/←.

69. У FTP-клієнті Far (Alt-F1/F2 → FTP) створити запис (Shift-F4) для роботи з FTP-сервером кафедри комп'ютерних та інформаційних систем, який на момент написання цих методичних вказівок знаходиться за адресою 192.168.3.252.

70. Спробувати з'єднатися з FTP-сервером. За необхідності перевірити відгук FTP-сервера на команду PING 192.168.3.252 і скоригувати налаштування з'єднання. У разі наявності з'єднання завантажити з FTP-сервера в каталог «Лабораторні роботи» файли будь-яких методичних матеріалів з навчальної дисципліни «Обчислювальна техніка».

71. Створити або додати (F2 → Insert) у головне меню користувача

пункти для запуску використовуваного Інтернет-браузера та медіаплеєра.

72. Створити в головному меню користувача підменю «Microsoft Office» і додати в нього команди для виклику встановлених офісних додатків (winword.exe, excel.exe і т.д.)

73. Створити в головному меню користувача підменю «Програмування» і додати у нього команди для:

а) запуску системи програмування Turbo Pascal 7.0 (файл turbo.exe);

б) запуску системи програмування Borland C++ (файл bc.exe);

в) запуску оболонки для програмування на асемблері RadAsm (файл RadAsm.exe).

74. Створити або додати в файл оброблення розширень (F9 → Команды → Ассоциации файлов) такі обробники для файлів указаних типів:

а) \*.pas – середовище програмування Turbo Pascal;

б) \*.bcc – середовище програмування Borland C++;

в) \*.rar – оболонка для програмування RadAsm.

75. Завершити всі додатки, натиснувши клавішу F10, завершити роботу з Far Manager і завершити сеанс роботи Windows.

### **Зміст звіту**

1. Назва та мета лабораторної роботи.

2. Результати, отримані під час виконання лабораторної роботи, і необхідні пояснення, як зазначено у відповідних пунктах робочого завдання.

3. Письмові відповіді на контрольні питання.

### **Контрольні питання**

1. Що являє собою Far Manager?

2. Яке призначення на основних файлових панелях Far Manager мають такі функціональні клавіші та їх комбінації:

а) F1 – F10;

б) Alt-F1 – Alt-F10;

в) Ctrl-F1 – Ctrl-F10;

г) Shift-F1 – Shift-F10?

3. Що таке протокол сеансу роботи з файловим менеджером?

**Література:** [4, с. 53–106, 9, с. 240–290].

## **Лабораторна робота № 2**

### **Тема. Сценарії командної оболонки Windows NT**

**Мета:** набуття практичних навичок розроблення простих сценаріїв в оболонці командного інтерпретатора ОС Windows NT.

### **Короткі теоретичні відомості**

Системні утиліти адміністрування з графічним інтерфейсом та файлові менеджери придатні, коли нам потрібно виконати будь-яку дію декілька разів, але для періодично повторюваних дій, та ще й масових, вони непридатні. Програми, засновані на діалозі людини з комп'ютером, є гарантованим джерелом помилки за досить великого обсягу введених даних. У такому разі слід перекласти всю рутинну та повторювану роботу на комп'ютер, що, як відомо, не думає, а тому не помиляється, дозволяє застосування командних файлів.

Командні файли використовуються в багатьох ОС, серед яких Windows, Mac OS й Linux-подібні системи. Запуск за розкладом (AT або SCHEDULE) попередньо розробленого командного файлу може використовуватися для резервного копіювання або архівації файлів, антивірусного сканування, очищення тимчасових каталогів і кошика, перевірки та дефрагментації твердого диска, вимикання комп'ютера в заданий час і т. д.

Командний файл у Windows – це звичайний текстовий файл із розширенням bat або cmd, у якому записані будь-які припустимі команди операційної системи (як зовнішні, так і внутрішні), а також деякі додаткові інструкції й ключові слова мови командної оболонки (command shell language), що роблять командні файли дещо подібними до алгоритмічних мов програмування. Файл повинен створюватися у *кодуванні OEM*. Загалом, кодування має значення тільки у разі, якщо в тексті використовуватимуться

символи національного алфавіту.

Командні файли також називають сценаріями, або скриптами (англ. script – сценарій) оболонки (shell script), або, ще по-старому, пакетними чи навіть bat-файлами (англ. batch – пакет). Для командного інтерпретатора Windows NT «рідним» розширенням є cmd, однак він також виконує й успадковані від попередніх версій ОС файли з розширенням bat. Розширення cmd є ознакою того, що командний файл може використовувати розширені можливості командного інтерпретатора Windows NT, які не підтримуються його попередниками в ОС DOS і Windows 9x/ME. Запуск на виконання командного файлу виконується так само, як і запуск програмних файлів з розширенням exe, а для консольних додатків Windows і зовнішньо має такий самий вигляд. Виконання командного файлу призводить до того самого результату, що й послідовне ручне введення записаних у ньому команд у командному рядку.

Мова командної оболонки є такою, що інтерпретується. Для виконання (інтерпретації) сценарію оболонки операційна система створює екземпляр процесу для командного інтерпретатора cmd.exe. Ніякої попередньої компіляції або перевірки синтаксису командного файлу не передбачається, і якщо під час його виконання зустрічається рядок з помилковою командою, то, зазвичай, вона просто ігнорується.

Мова командної оболонки Windows надає доволі гнучкі можливості створення сценаріїв, подібні до засобів алгоритмічних мов програмування. Синтаксис багатьох конструкцій мови командної оболонки наближений до аналогічних засобів мови C/C++. Вона використовує змінні, має набір арифметичних і логічних операторів, операторів виведення, циклів, умовного та безумовного передавання керування, і з її використанням можливе написання не тільки маленьких лінійних послідовностей команд, а і складних сценаріїв з умовними розгалуженнями та циклами.

Окрім будь-яких внутрішніх або зовнішніх команд, які в мові командної оболонки можна вважати аналогом бібліотечних процедур і функцій алгоритмічних мов програмування, у командних файлах використовуються і

специфічні тільки для них команди, до яких належать CALL, ECHO, ENDLOCAL, FOR, GOTO, IF, PAUSE, REM, SETLOCAL, SHIFT. Серед цих команд у звичайному сеансі командного рядка привертає увагу хіба що використання команди FOR. Використання інших команд у командному рядку хоча і припустимо, але просто не має сенсу.

Навести опис призначення та синтаксису використання всіх названих команд у цих методичних вказівках немає ні можливості, ні потреби. За необхідності завжди можна звернутися до довідкової системи Windows. Довідку щодо будь-якої команди можна також отримати безпосередньо з командного рядка за допомогою звичного параметра /?. Далі ми на прикладах розглянемо деякі можливості, що надають командні файли, з невеличкими поясненнями.

Наприклад, існує доволі велика кількість програм, які дозволяють виконувати очищення магнітних дисків, видаляючи непотрібні файли – тимчасові, резервні копії або будь-які інші, визначені користувачем. Однак для людини, знайомої з можливостями командного рядка, а тим більше – з командними файлами, такі програми навряд чи будуть цікаві. Усього одна виконана з командного рядка команда:

```
for /R C:\ %f in (*.bak,*.tmp) do del /f /q %f
```

виконує видалення файлів з розширеннями bak і tmp на *всьому* диску C:, у тому числі файли з атрибутом «тільки для читання» без запиту на підтвердження видалення.

Замість змінної циклу, позначеної %f, командний інтерпретатор спочатку підставляє імена всіх файлів, що задовольняють маску \*.bak, а потім – \*.tmp. Пошук таких файлів починається з кореневого каталогу диска C:, а параметр /R оператора FOR робить цей пошук рекурсивним, тобто переглядаються всі підкаталоги кореневого каталогу диска C: разом з їх підкаталогами, підкаталогами їх підкаталогів і т. д.

З урахуванням того, що маску пошуку файлів у команді можна значно розширити, однак якщо аналогічну задачу потрібно розв'язувати ще і для диска

D:, то доведеться вводити в командному рядку вже дві схожі команди. Ці дві команди можуть утворити командний файл, а як діяти, якщо наперед невідомо, скільки логічних дисків у системі?

Універсальним розв'язанням такої задачі може бути такий командний файл:

```
@echo off
for %%d IN (c d e f g h i j k l m n o p q r s t u v w x y z) do (
    if not exist %%d:\ exit /b
    pushd %%d:\
    for /R %%f in (*.bak,*.tmp) do del /f /q %%f
    popd
)
```

Команда `echo off` призначена для заборони виведення на консоль виконуваних команд, а оператор `@` перед командою притримує виведення на консоль самої команди `echo off` і командного рядка. Оператор `@` можна використовувати перед будь-якою командою, а команди `echo off` та `echo on` можна використовувати у файлі декілька разів, вимикаючи та вмикаючи `echo`-повторення виконуваних команд. Перший рядок з командою `@echo off` є фактичним стандартом для багатьох командних файлів.

Одразу слід зазначити ще одне корисне використання команди `echo`. Команда `echo` є стандартним оператором виведення мови командної оболонки, й режим `echo off` чи `echo on` не позначиться на дії самої команди. Використання стандартних операторів для виведення проміжних результатів роботи програми завжди вважалось найпростішим засобом налагодження. Помилки у командному файлі можуть пошкодити систему. Для розроблення командного файлу замість «небезпечних» команд (`del`, `rd`, `move` тощо), спочатку краще використовувати команду `echo`. Лише переконавшись, що сценарій працює належно, `echo` можна замінити на потрібну команду.

Зовнішній цикл `for` використовується для того, щоб надати по чергово змінній `%%d` значення `c`, `d`, `e` і так далі. Команда `if not exist %%d:\ exit /b`

використовується для перевірки існування в системі логічного диска з іменем %%d і буквально означає «якщо (if) не (not) існує (exist) кореневий каталог (\) диска %%d:, то вийти (exit)», тобто завершити роботу командного файлу. Якщо умова не виконується, тобто існує кореневий каталог наступного диска, команда pushd %%d:\ робить його поточним, у внутрішньому циклі for скороченої форми виконується рекурсивне видалення \*.bak і \*.tmp від поточного каталогу, а popd поновлює вихідний поточний каталог для командного файлу.

Розглянемо такий приклад. Програміст щоденно займається розробкою програмного забезпечення. Усі каталоги розроблюваних ним програм завжди розміщуються у підкаталозі «Мої проекти» каталогу «Мої документи». Він хоче організувати щоденне резервне копіювання розроблюваних ним проектів і висуває такі вимоги.

По-перше, копіюватися повинні тільки вихідні файли проектів. Немає потреби копіювати програмні та допоміжні файли, які генеруються під час компіляції проектів, оскільки, маючи вихідні, їх завжди можна легко отримати.

По-друге, щоденному резервному копіюванню підлягають не всі вихідні файли, а тільки ті, які були змінені протягом того самого дня, у який виконується резервне копіювання.

Окрім того, вихідні файли проектів повинні копіюватися разом зі структурою каталогів, у яких вони розміщуються, оскільки це важливо для компіляції проектів.

По-третє, резервне копіювання повинно виконуватися у зазначене місце та у каталог, назва якого містила б дату виконання операції.

Організувати таке резервне копіювання можна, запланувавши на виконання командний файл такого вмісту:

```
@echo off
set today=%date:~6,4%-%date:~3,2%-%date:~0,2%
set upday=%date:~3,2%-%date:~0,2%-%date:~6,4%
set src=%USERPROFILE%\Мої документи\Мої проекти
```

```

set bak=F:\Backup\backup%today%
set log=%bak%\backup%today%.log
rem set log= F:\Backup\backup%today%\backup%today%.log
if not exist "%bak%" md "%bak%"
pushd "%src%" ||exit /b 1
for /r %%f in (*.obj,*.exe,*.dll,*.res,*.sys,*.bak,*.pdb,*.exp,*.lst,*.map,*.sbr)
do ^
del /F /Q "%%f">>"%log%"
popd
xcopy "%src%" "%bak%" /S /D:%upday% /V /C /F /R /Y /I>>"%log%"

```

Рядки з командою `set` демонструють використання у командних файлах змінних. Певним чином можна сприймати команду `set` як оператор присвоєння алгоритмічних мов. Наприклад, аналогом `a:=5` для Pascal є `set a=5` мовою командного інтерпретатора. Відразу слід попередити, що така аналогія дуже умовна. Командний інтерпретатор не розрізняє типів і працює зі змінними як із набором символів, тобто з текстом. Цим він більше схожий на макрогенератор, або препроцесор, тих мов програмування, які підтримують макрозасоби. Щодо оброблення більш наближеним аналогом для `set a=5` для Pascal є навіть `const a=5`, але *змінні* командного інтерпретатора, на відміну від констант Pascal, можна *змінювати* скільки завгодно далі у сценарії після їх опису.

Змінна `today` призначена для формування коротшого імені рядка, що містить запис (текст!) поточної дати. Системна змінна оточення `%date%` у такій локалізації надає дату у форматі `dd.mm.yyyy`, наприклад, `31.12.2009`. Це виявляється незручним для використання дати як імені або частини імені файлу чи каталогу. Під час сортування за іменем таких файлів/каталогів порушується хронологічний порядок, оскільки, наприклад, `31.12.2008` і `31.12.2009` відобразатимуться поряд, незважаючи на різницю між ними у цілий рік. Зручним щодо цього є формат `yyyy.mm.dd`, або, наприклад, `yyyy-mm-dd`, який використовується у нашому прикладі.



Запис `set today=%date:~6,4%-%date:~3,2%-%date:~0,2%` для командного інтерпретатора означає, що він повинен спочатку із запису (рядка тексту!) змінної оточення `%date%` узяти під рядок з чотирьох символів, починаючи із шостої позиції (нумерація з нуля), потім дописати справа до отриманого підрядка символ «-», потім дописати справа до отриманого вже таким чином підрядка ще два символи, узяті зі змінної оточення `%date%`, починаючи з третьої позиції і т. д.

Запис `set upday=%date:~3,2%-%date:~0,2%-%date:~6,4%` розв'язує аналогічну задачу й призначений для присвоювання змінній `upday` значення поточної дати у форматі `mm-dd-yyuu`. Саме у такому форматі повинна задаватися дата файлів, що підлягають копіюванню у параметрі `/D:mm-dd-yyuu` команди `xcopy`.

Змінні `src`, `bak` та `log` визначають місце розташування каталогу, що підлягає копіюванню, місце розташування та ім'я каталогу призначення та ім'я `log`-файлу, у якому будуть записані імена всіх скопійованих файлів із зазначенням їх попереднього місця розташування. Сам `log`-файл буде збережено у каталозі призначення. Для звертання до значення змінної використовуються вже відомі нам «лапки» із символів «%».

Будь-який рядок командного файлу, що починається з `rem`, не обробляється командним інтерпретатором і зазвичай містить коментарі. У нашому командному файлі рядок коментаря демонструє еквівалентне визначення значення змінної `log`. Таке визначення, можливо, здається і зрозумілішим, але виявляється незручним. Незручність полягає у тому, що якщо потім із часом доведеться адаптувати цей командний файл через зміну розташування каталогу призначення, то будемо вимушені змінювати два рядки замість одного.

Рядок з коментарем також демонструє зручність використання змінних. Значення `src` і `bak` ми використовуємо декілька разів, і з їх безпосереднім зазначенням у командному файлі його модернізація потребувала б уже значних змін, а не зміни тільки двох рядків з визначеннями відповідних змінних на

самому початку командного файлу.

Оператор `if not exist "%bak%" md "%bak%"` призначений для створення вихідного каталогу. Далі за допомогою зв'язки `pushd "%src%" ||exit /b 1` виконується спроба переходу у вихідний каталог. Якщо він не існує, то буде виконано команду `exit /b 1`, яка завершить виконання командного файлу з кодом повернення (`errorlevel`), що дорівнює 1. Використання зв'язки, яка призводить до умовного виконання команд, тут дуже важливе. Незавжно передбачити наслідки виконання наступного циклу `for`, якщо спроба змінити поточний каталог на каталог призначення виявиться невдалою, а поточним буде, наприклад, каталог `%ProgramFiles%` або `%SystemRoot%`.

Після видалення непотрібних файлів за допомогою знайомої вже нам форми використання оператора `for` копіювання інших файлів виконується за допомогою команди `xcopy` (див. довідку щодо команди). Символ «`^`» в операторі `for` зокрема і командних файлах узагалі може використовуватися для перенесення тексту на наступний рядок без *логічного* розриву рядка. Символ «`^`» у командних файлах використовується також для так званого «екранування». Він повинен передувати будь-якому зі спецсимволів, якщо можлива неправильна інтерпретація цього спецсимволу в контексті командного файлу.

У наступному прикладі розширимо функціонал нашого попереднього командного файлу завдяки можливості використання додаткових опцій у командному рядку під час його виклику. Користувач зможе самостійно задавати вихідний каталог і каталог-приймач і, за необхідності, відмовитися від видалення бінарних файлів. Після незначного опрацювання командний файл матиме такий вигляд:

```
@echo off
set today=%date:~6,4%-%date:~3,2%-%date:~0,2%
set upday=%date:~3,2%-%date:~0,2%-%date:~6,4%
if {%1}=={ } goto help
if {%2}=={ } goto help
```

```

set src=%~f1%
set bak=%~f2\backup%today%
set log=%bak%\backup.log
if not exist "%bak%" md "%bak%"
if /i {%3}=={/b} goto skip
pushd "%src%" ||exit /b 1
for /r %%f in (*.obj,*.exe,*.dll,*.res,*.sys,*.bak,*.pdb,*.exp,*.lst,*.map,*.sbr)
do ^
del /F /Q "%%f"
popd
:skip
xcopy "%src%" "%bak%" /S /D:%upday% /V /C /F /R /Y /I>>"%log%"
exit /b 0
:help
echo Виконує резервне копіювання каталогу ^<джерело^>
echo в каталог ^<приймач^>\backup%today%
echo.
echo Використання:
echo %~nx0 [[диск:\[шлях\]]джерело [[диск:\]шлях\]приймач [/b^/B]
echo /b або /B – не виконувати видалення бінарних файлів

```

У командні файли можна передавати параметри командного рядка майже так само, як це робиться для будь-яких внутрішніх або зовнішніх команд. Ознакою параметра в командному рядку є символ пробілу, тобто командний інтерпретатор вважає за параметр будь-яке *слово тексту*. За потреби передати у командний файл параметр, що містить пробіл, наприклад, C:\Program Files, він повинен бути взятий у лапки – "C:\Program Files".

Деякі внутрішні або зовнішні команди дозволяють не розділяти параметри пробілами. Наприклад, `dir /b /d`, `dir /b/d` та `dir/b/d` виконуються однаково, а для командного файлу для передавання двох параметрів «/b» і «/d» правильним варіантом буде перший. Другий варіант припустимий, але у цьому

випадку в командний файл буде передано єдиний параметр «/b/d» і доведеться самотійно виконувати його додаткову обробку засобами командного файлу.

Для звертання до параметрів у командному файлі використовуються їх номери від %0 до %9. Значення параметра %0 дорівнює повному імені командного файлу. Використанням команди SHIFT дозволяє обробити в командному файлі більше ніж 10 параметрів.

Оператори `if {%1}=={ } goto help` та `if {%2}=={ } goto help` за відсутності першого та другого параметрів передають керування на мітку `help`, з якої починається виведення на екран підказки щодо призначення та способу використання цього командного файлу. Замість дужок «{ }» можна використовувати будь-які два, а то і один символ, наприклад, `if -%1==-`. Просто за синтаксичними вимогами `if` після знака порівняння `==` хочь щось повинно бути. За відсутності параметра умова перетворюється на `{ }=={ }`, тобто істину.

Команда `set src=%~f1%` використовується для призначення місця розташування вихідного каталогу. Конструкція виду `%~f1%` у такому разі розширює ім'я вихідного каталогу, яке передається параметром командного рядка %1 до повного імені. Інші варіанти розширень параметрів можна подивитися у довідці, наприклад, за командою FOR. Загалом розширення припускається для будь-якого параметра командного рядка і навіть узагалі для будь-якого параметра командного файлу, якщо він означає файл чи каталог. Команда `set bak=%~f2\backup%today%` виконує аналогічні дії для каталогу приймача.

Оператор `if /i {%3}=={/b} goto skip` передає керування на мітку `skip`, якщо у третьому параметрі командного рядка був зазначений параметр /b або /B (але не b, B, -b, -B, --b, \B і т. і.). Регістрозалежне порівняння виконується завдяки опції /i оператора `if`. Перехід на мітку `skip` призводить до невиконання частини команд сценарію, які видаляють бінарні файли з вихідного каталогу. Після команди `xcopy`, яка виконує копіювання за повними іменами каталогів, команда `exit /b 0` завершує виконання командного файлу з кодом повернення (`errorlevel`) 0 для того, щоб не виконувати команди, які починаються з мітки `help`.

У командах, що починаються з мітки help, заслуговує уваги використання символу «^» для екранування символів «<», «>» та «|», які можуть сприйматися командним інтерпретатором як перенаправлення введення/виведення і конвеєризація, використання echo. (крапка обов'язкова) для пропуску під час виведення одного рядка, та розширення %~nx0 параметра %0 до імені й розширення командного файлу.

### **Хід роботи**

1. Увімкнути комп'ютер, завантажити операційну систему Windows NT, увійти до свого облікового запису і виконати запуск Far Manager.

2. Перейти до каталогу «%USERPROFILE% \ Мои документы \ Моя освіта \ Університет \ Навчання \ 1-й курс \ 1-й семестр \ Обчислювальна техніка \ Лабораторні роботи \ WINCMD».

3. Використовуючи параметр /? у командному рядку або звернувшись до послуг довідкової системи, вивчити призначення та синтаксис використання команд CALL, ECHO, ENDLOCAL, FOR, GOTO, IF, PAUSE, REM, SETLOCAL, SHIFT.

4. Створити у поточному каталозі командний файл viewlog.cmd, який виводить посторінково на екран монітора вміст файлів з розширеннями \*.log й \*.txt у заданому викликом командному рядку каталозі. Синтаксис виклику viewlog.cmd [диск:\шлях]. Квадратні дужки «[]» тут і далі позначають необов'язковість узятого у них параметра. Якщо параметр командного рядка диск:\шлях не заданий, то за замовчанням використовується поточний каталог.

5. Створити у поточному каталозі командний файл delbak.cmd, який видаляє всі файли з розширенням \*.bak у заданому в командному рядку каталозі, включаючи файли з установленими атрибутами «прихований», «системний», «тільки для читання». Синтаксис виклику delbak.cmd [диск:\шлях] [/y]. З указанням опції /y видалення провадиться без запиту підтвердження користувача. Якщо параметр командного рядка не заданий, то за замовчанням використовується поточний каталог.

6. Створити у поточному каталозі командний файл deltmp.cmd, який

видаляє всі файли й каталоги в тимчасовому каталозі користувача, заданому змінною оточення %TEMP%, включаючи файли та каталоги з установленими атрибутами «прихований», «системний», «тільки для читання». Синтаксис виклику `deltmp.cmd [/u]`. При вказівці опції `/u` видалення провадиться без запиту підтвердження користувача.

7. Створити у поточному каталозі командний файл `ff.cmd`, який шукає місце розташування заданого файлу в заданому каталозі й у всіх його підкаталогах і виводить на консоль повну інформацію (атрибути, дата, час, розмір) про знайдені файли. Синтаксис виклику `ff.cmd файл [диск:\шлях]`. Перший параметр є обов'язковим. Якщо другий параметр `диск:\шлях` не заданий, то пошук починається з поточного каталогу. Пошук повинен виконуватися для файлу з будь-якими атрибутами.

8. Створити у поточному каталозі командний файл `srext.cmd`, який копіює в каталог призначення будь-яку кількість зазначених файлів. Синтаксис виклику `srext.cmd диск:\шлях\призначення диск:\шлях\файл1 [диск:\шлях\файл2] [...]`. Перший параметр, що задає каталог призначення, є обов'язковим. При заданні імен файлів можна використовувати символи «\*» й «?».

9. Створити у поточному каталозі командний файл `sync.cmd`, який виконує синхронізацію каталогів за вмістом. Синтаксис виклику `sync.cmd [диск:\шлях1] диск:\шлях2`. Якщо файли, розташовані в `диск:\шлях1`, відсутні в `диск:\шлях2`, то вони копіюються в `диск:\шлях2`, і навпаки, якщо файли, розташовані в `диск:\шлях2`, відсутні в `диск:\шлях1`, то вони копіюються в `диск:\шлях1`. Якщо `диск:\шлях1` не заданий, то за замовчанням використовується поточний каталог.

10. Створити у поточному каталозі командний файл `delcopy.cmd`, який виконує синхронізацію каталогів за вмістом. Синтаксис виклику `delcopy.cmd [диск:\шлях1] диск:\шлях2`. Командний файл повинен видалити всі файли в `диск:\шлях2`, які є в `диск:\шлях1`. Якщо `диск:\шлях2` незаданий, то за замовчанням використовується поточний каталог.

11. Створити у поточному каталозі командний файл `cmp.cmd`, який порівнює два задані каталоги за набором файлів, а однойменні файли – за вмістом. Синтаксис виклику `cmp.cmd [диск:\шлях1] диск:\шлях2`. Якщо перший параметр не заданий, то за замовчанням використовується поточний каталог.

12. Перевірити працездатність усіх розроблених командних файлів для різних наборів параметрів і, за необхідності, виконати їх налагодження.

### Зміст звіту

1. Назва й мета лабораторної роботи.
2. Вміст усіх розроблених командних файлів з описом призначення і синтаксису виклику.
3. Письмові відповіді на контрольні питання.

### Контрольні питання

1. Яка різниця між командами `for %%f in (*.bak) do del /f /q %%f`, `for /R %%f in (*.bak) do del /f /q %%f` і `for /R C:\ %%f in (*.bak,*.tmp) do del /f /q %%f`?

2. Що відбудеться з параметрами `%0-%9` командного файлу після виконання у ньому команди `shift /3`?

3. У каталозі `C:\Windows` розташований файл `batch.cmd` розміром 66 байтів. Останнє редагування файлу виконувалось 31.12.2008 о 23.55. Після цього файлу було встановлено атрибути «прихований» і «тільки для читання». У тексті цього файлу використовуються комбінації символів `%~0`, `%~f0`, `%~d0`, `%~p0`, `%~n0`, `%~x0`, `%~z0`, `%~a0`, `%~t0`, `%~z0`, `%~dp0`, `%~nx0`, `%~ftza0`. Що вони позначають? Що буде відображено на консолі в результаті виконання команди `echo %~nxtz0`?

4. Для чого використовується форма оператора `if defined` змінна команда `[else ...]`? Запропонувати приклади використання.

5. Для чого використовується форма оператора `if [not] errorlevel` число команда `[else ...]`? Запропонувати приклади використання.

**Література:** [4, с. 331–336, 7, с. 30–66, 9, с. 347–354].

## Лабораторна робота № 3

### Тема. Командний інтерпретатор Linux

**Мета:** набуття практичних навичок одержання інформації про систему, використання довідкової служби та виконання основних файлових операцій у командній оболонці ОС Linux.

### Короткі теоретичні відомості

Для роботи з командним інтерпретатором ОС Linux повною мірою знадобиться досвід і практичні навички, набуті під час виконання лабораторних робіт, присвячених командному інтерпретатору Windows NT, хоча, звичайно, є і певні відмінності.

У Linux не існує поняття логічного диска. Для гнучких магнітних дисків, розділів твердих магнітних дисків, CD/DVD-ROM, електронних RAM-дисків, накопичувачів на EEPROM (Flash) і взагалі для будь-яких пристроїв використовується файлова абстракція. Будь-який пристрій відображається як файл або підкаталог системного каталогу /dev, наприклад, /dev/hda2 – другий (2) розділ на першому (a) твердому диску (hd). Змінні носії відображаються як файли або підкаталоги системного каталогу /media, наприклад, /media/cdrom0 – перший CD/DVD-ROM. Далі за допомогою так званої «операції монтування» пристрій може *відображатися* як підкаталог будь-якого каталогу (точки монтування).

У Linux не існує поняття каталогу встановлення та системного каталогу, аналогічних змінним Windows NT %SystemRoot% та %SystemRoot%\System32. Для користувача верхнім рівнем ієрархічної структури, що використовується для організації зберігання даних, є кореневий каталог з псевдоіменем «/» на який *монтуються* всі інші системні каталоги /bin, /boot, /dev, /etc, /home, /lib, /lost+found, /media, /mnt, /opt, /proc, /root, /sbin, /tmp, /user, /var і деякі інші *незалежно від їх фізичного розташування*.

У Linux також існують змінні оточення, для позначення яких використовується символ «\$», що передує імені змінної. Аналогами



%USERPROFILE% та %PATH% Windows NT у Linux виступає \$HOME та \$PATH.

Імена файлів/каталогів у Linux є *регістрозалежними*: File, file, FILE – це різні імена файлів/каталогів, що можуть знаходитися в одному і тому самому підкаталозі. У Linux не використовується розширення імені файлу/каталогу і символ крапки «.» є звичайним символом імені, окрім єдиного випадку. Якщо ім'я файлу/каталогу *починається* із символу крапки «.», то такий файл/каталог вважається прихованим і деякі команди його «не бачать».

Імена файлів/каталогів із символами пробілів повинні використовуватися у лапках «». Є певний набір символів, які не *рекомендується* (але і не забороняється) використовувати в іменах файлів/каталогів, оскільки, вони мають спеціальне призначення для командного інтерпретатора й можуть ним сприйматися не однозначно. До таких символів належать, зокрема «|», «>», «&», «;» які в Linux також використовуються для перенаправлення введення/виведення і зв'язки команд. У Linux також використовується команда-фільтр more, однак команда find має інше призначення й аналогом пари команд find та findstr Windows NT виступає єдина (і потужна) команда-фільтр grep.

У роботі з файлами в Linux також широко використовуються символи підстановки (шаблони або джокери) «\*» й «?», які означають те саме, що і у Windows NT. Використовуючи ці символи, слід пам'ятати про регістрозалежність імен і те, що в Linux немає розширень: \*.\* у Linux означає не «усі», а ті, що мають символ крапки в імені, а\* та A\* – різні.

Забороненим в іменах файлів/каталогів можна вважати єдиний символ «/» який використовується як роздільник під час запису шляху до файлу/каталогу. Повні, часткові та відносні імена файлів/каталогів у Linux виписують аналогічно тому, як це робиться і у Windows NT, але через прямий слеш «/», з якого завжди починається повний шлях.

У Linux атрибути файлів/каталогів використовуються у дещо іншому розумінні, ніж у Windows NT. Як уже зазначалося, файли/каталоги, імена яких починаються із символу крапки, *уважаються* прихованими, але в Linux немає

такого атрибута. Об'єкт файлової системи Linux має десять атрибутів, що визначають його тип і права доступу до нього, для позначення яких використовується символічний рядок виду `t gwx gwx gwx`.

Перший символ позначений як «t», визначає тип об'єкта: «-» – файл, «d» – каталог, «b» – блоковий пристрій, «c» – символічний пристрій, «l» – символічний зв'язок, «p» – іменованний канал, «s» – гніздо.

Наступні дев'ять символів розділено на три групи, що визначають права користувача (User), учасників групи (Group), до якої належить користувач, і всіх інших (Other). У межах кожної групи символи `gwx` означають одне і те саме: `r` – право на читання, `w` – право на запис, `x` – право на виконання. Відсутність певного права позначається символом «-» у відповідній позиції. Наприклад, рядок виду `-rw-r-----` означає, що користувач (господар) файлу має право на читання і запис цього файлу, учасники групи користувача мають право на читання цього файлу, будь-кому іншому доступ до файлу заборонений. Право на читання і виконання відносно каталогу означає можливість переглядати вміст і переходити у цей каталог.

### **Хід роботи**

1. Увімкнути комп'ютер, завантажити операційну систему Linux, увійти до свого облікового запису й відкрити сеанс роботи командного інтерпретатора ОС.

2. За допомогою команд `uname -srv` та `uname -mri` визначити й занести до звіту версію операційної системи та апаратної платформи.

3. За допомогою команди `echo $SHELL` визначити й занести до звіту тип використовуваної командної оболонки.

4. За допомогою команд `pwd` та `echo $HOME` визначити і занести до звіту місце розташування свого поточного й домашнього каталогу.

5. За допомогою команди `echo $PATH` визначити і занести до звіту каталоги пошуку програм.

6. За допомогою команди `man` команда (`man 1 команда`) або опції `--help` ознайомитися з призначенням і опціями команд `ls`, `mkdir`, `rmdir`, `cat`, `cp`, `mv`, `rm`,

find, grep, ln, chmod.

7. Використовуючи символи перенаправлення, вивести у файли у своєму початковому каталозі довідку за командами mkdir, rmdir, cat, cp, mv, rm, find, grep, ln, chmod. Як ім'я кожного файлу використати ім'я відповідної команди з «розширенням» hlp.

8. Створити у своєму початковому каталозі підкаталоги Help, Script, Common, Recycled і каталог Mail з підкаталогами Inputbox й Outputbox.

9. Створити в каталозі Help підкаталоги Commands й System, а в каталозі Recycled – підкаталоги Temp й Bak.

10. Скопіювати всі отримані в п. 7 довідкові (hlp-) файли в каталог Commands.

11. Створити в початковому каталозі копії отриманих у п. 7 довідкових hlp-файлів, додавши до їх імен «друге розширення» .bak.

12. Перенести отримані у попередньому пункті bak-копії hlp-файлів у каталог Bak каталогу Recycled.

13. Створити у своєму початковому каталозі повну копію каталогу Help з ім'ям Files.

14. Скопіювати каталог Help разом з підкаталогами в підкаталог Temp каталогу Recycled.

15. Перенести каталог Files разом з підкаталогами в каталог Recycled з новим ім'ям TEMP.

16. Видалити каталог TEMP.

17. Видалити всі файли й каталоги в каталозі Recycled.

18. Знайти у своєму початковому каталозі та його підкаталогах усі файли, в іменах яких є слово «find».

19. Знайти у своєму початковому каталозі та його підкаталогах усі файли, що містять слово «find».

20. Створити прихований каталог з ім'ям Bak у початковому каталозі.

21. Створити у каталозі Bak початкового каталогу жорсткий зв'язок з ім'ям hard\_link\_ln.hlp з файлом ln.hlp початкового каталогу.

22. Створити в каталозі Вак початкового каталогу м'який (символічний) зв'язок з ім'ям `symlolic_link_chmod.hlp` з файлом `chmod.hlp` початкового каталогу.

23. Додати будь-який рядок у файл `ln.hlp` і переглянути вміст файлу `hard_link_ln.hlp`.

24. Додати будь-який рядок у файл `hard_link_ln.hlp` і переглянути вміст файлу `ln.hlp`.

25. Видалити файл `ln.hlp` і переглянути вміст `hard_link_ln.hlp`.

26. Додати рядок у файл `chmod.hlp` і переглянути вміст файлу `symlolic_link_chmod.hlp`.

27. Додати рядок у файл `symlolic_link_chmod.hlp` і переглянути вміст файлу `chmod.hlp`.

28. Видалити файл `chmod.hlp` і переглянути вміст `symlolic_link_chmod.hlp`.

29. Видалити всі файли в каталозі Вак та всі `hlp`-файли в початковому каталозі.

30. Установити для всіх файлів і каталогів свого початкового каталогу права на читання й запис тільки для власника.

31. Дозволити для каталогу `Common` читання та виконання для всіх користувачів.

32. Створити в каталозі `Common` файл `chat` із текстом «Дошка оголошень».

33. Установити для файлу `chat` права на читання й запис для всіх користувачів.

34. Закрити консоль командного інтерпретатора й завершити сеанс роботи.

### **Зміст звіту**

1. Назва та мета лабораторної роботи.

2. Результати, отримані під час виконання лабораторної роботи, та необхідні пояснення, як зазначено у відповідних пунктах робочого завдання.

3. Письмові відповіді на контрольні питання.

### **Контрольні питання**

1. Що виконує команда `man`? Перелічити розділи і команди керування `manual pages`.
2. Які типи файлових об'єктів існують у Linux?
3. Що робить команда: `ls -la /usr/bin|grep “^.....x”|more`?
4. Що роблять команди `chmod 664 file` та `chmod 700 dir`, якщо `file` – це файл, а `dir` – каталог?
5. Чим подібні та відмінні жорсткі та символічні зв'язки?
6. Як у Linux можна організувати обмін файлами між користувачами групи?

**Література:** [1, с. 96–112, 2, с. 129–147, 3, с. 71–146, 5, с. 56–80, 6, с. 79, 83, 99, 140, 298].

### **Лабораторна робота № 4**

**Тема. Файловий менеджер GNU Midnight Commander і сценарії оболонки Linux**

**Мета:** набуття практичних навичок роботи з файловим менеджером Midnight Commander і розроблення простих сценаріїв командної оболонки Bourne Again Shell (bash) ОС Linux.

#### **Короткі теоретичні відомості**

Midnight Commander – один з популярних консольних файлових менеджерів для Unix-подібних операційних систем, ОС Microsoft Windows, OS/2 і Mac OS X. Midnight Commander зараз включений у більшість дистрибутивів Linux і поширюється під GNU General Public License. У більшості дистрибутивів програма Midnight Commander не встановлюється автоматично під час інсталяції системи, але відповідний rpm-пакет, зазвичай, є на дистрибутивному диску.

Midnight Commander підтримує всі основні операції з файлами: копіювання, переміщення, архівація, зміна прав доступу, створення посилань, каталогів та інші. Midnight Commander може працювати з файлами на віддалених файлових системах за допомогою Samba, FTP й SSH. У Midnight Commander вбудований текстовий редактор файлів з підсвічуванням синтаксису мов програмування, здатний працювати у режимах ASCII та HEX з підтримкою регулярних виразів. Цікавою особливістю є також можливість використання регулярних виразів при виконанні операцій у файлових панелях.

Для роботи з Midnight Commander в ОС Linux повною мірою нагодиться досвід і практичні навички, набуті під час виконання лабораторних робіт, присвячених FAR Manager у середовищі Windows NT, – обидва підтримують двопанельну концепцію, мають схожий інтерфейс, у них використовуються схожі прийоми роботи, набори основних функціональних клавіш збігаються.

### **Хід роботи**

1. Увімкнути комп'ютер, завантажити операційну систему Linux, увійти до свого облікового запису й стартувати на виконання Midnight Commander командою `mc`.

2. Натиснувши клавішу F9, активізувати меню і виконати налаштування параметрів Midnight Commander за допомогою пункту меню Налаштування\Конфігурація...\  
:

- а) показ резервних і прихованих файлів;
- б) відключити відображення прихованих файлів і каталогів у панелях;
- в) переміщення курсора вниз при позначці файлів за допомогою клавіші Insert;
- г) ніколи не робити паузу після виконання команди;
- д) увімкнути показ деталей операцій і підрахунок загального розміру;
- е) підключити вбудований редактор і вбудовану програму перегляду файлів;
- ж) увімкнути режим зміни каталогів за посиланнями та безпечне видалення.

3. За допомогою [F9] \ Настроювання \ Зовнішній вигляд...\ установити такі настроювання панелей Midnight Commander:

- а) вертикальне розташування панелей рівного розміру;
- б) кольорове позначення різних типів файлів і прав доступу до них;
- в) відображення рядка підказки про призначення функціональних клавіш, рядка міні-статусу й командного рядка.

4. За допомогою [F9] \ Настроювання \ Підтвердження...\ увімкнути режим запиту підтвердження перед видаленням і перезаписом файлів.

5. За допомогою [F9] \ Права (ліва) панель \ Формат списку...\ установити скорочений формат списку файлів на лівій і правій панелях.

6. За допомогою [F9]\Права (ліва) панель \ Порядок сортування...\ установити в обох панелях режим сортування імен файлів за абеткою з урахуванням регістра символів.

7. Видалити (F8) у своєму робочому каталозі всі файли й каталоги, створені під час виконання попередньої лабораторної роботи.

8. Одержати (створити) у своєму початковому каталозі файли, що містять коротку довідку про призначення, параметри й опції команд `mkdir`, `rmdir`, `cat`, `cp`, `mv`, `rm`, `find`, `grep`, `ln`, `chmod`. Як ім'я кожного файлу використати ім'я відповідної команди з «розширенням» `hlp`.

9. Переглянути (F3) вміст отриманих довідкових файлів за допомогою вбудованої програми Midnight Commander.

10. Створити (F7) у своєму початковому каталозі підкаталоги `Help`, `Script`, `Common`, `Recycled` і каталог `Mail` з підкаталогами `Inputbox` й `Outputbox`.

11. Створити (F7) у каталозі `Help` підкаталоги `Commands` й `System`, а в каталозі `Recycled` – підкаталоги `Temp` й `Back`.

12. Скопіювати (F5) усі отримані в п. 8 довідкові (`hlp-`) файли в каталог `Commands`.

13. Створити (F5) в початковому каталозі копії отриманих у п. 8 довідкових файлів `ln.hlp` і `chmod.hlp`, додавши до їхніх імен «друге розширення» `.bak`.

14. Перенести (F6) отримані .bak–копії довідкових файлів у каталог Вак каталогу Recycled.

15. Порівняти вміст (Ctrl-X+D або F9\Команда) початкового каталогу й каталогу Commands.

16. Повторно одержати в початковому каталозі довідковий файл для команди chmod і після цього порівняти вміст початкового каталогу й каталогу Commands.

17. Додати (F4) в кінець файлу chmod.hlp початкового каталогу рядок із 78 символів «\*» і після цього порівняти вміст початкового каталогу й каталогу Commands.

18. Створити (F5) у своєму початковому каталозі повну копію каталогу Help з ім'ям Files.

19. Переконалися в повній ідентичності Help й Files.

20. Скопіювати (F5) каталог Help разом з підкаталогами в каталог Temp.

21. Перенести (F6) каталог Files разом з підкаталогами в каталог Recycled з новим ім'ям TEMP.

22. Порівняти каталоги Temp й TEMP.

23. Видалити (F8) каталог TEMP.

24. Видалити (F8) усе в каталозі Recycled.

25. Знайти (F9\Команда\Пошук файлу) у своєму початковому каталозі та його підкаталогах усі файли, в іменах яких є слово «find».

26. Знайти у своєму початковому каталозі та його підкаталогах усі файли, що містять слово «find».

27. Створити у початковому каталозі прихований каталог з ім'ям Вак.

28. Створити в каталозі Вак жорсткий зв'язок (F9\Файл) з ім'ям hard\_link\_ln.hlp з файлом ln.hlp початкового каталогу.

29. Створити в каталозі Вак жорсткий зв'язок (F9\Файл) з ім'ям symbolic\_link\_chmod.hlp з файлом chmod.hlp початкового каталогу.

30. Додати рядок у файл ln.hlp і переглянути вміст файлу hard\_link\_ln.hlp.

31. Додати рядок у файл hard\_link\_ln.hlp і переглянути вміст файлу ln.hlp.



32. Видалити файл ln.hlp і переглянути вміст hard\_link\_ln.hlp.

33. Додати рядок у файл chmod.hlp і переглянути вміст файлу symbolic\_link\_chmod.hlp.

34. Додати рядок у файл symbolic\_link\_chmod.hlp і переглянути вміст файлу chmod.hlp.

35. Видалити файл chmod.hlp і переглянути вміст symbolic\_link\_chmod.hlp.

36. Видалити все в каталозі Bak.

37. Видалити всі hlp-файли в початковому каталозі.

38. Установити (F9\Файл) для всіх файлів і каталогів свого початкового каталогу права на читання й запис тільки для власника.

39. Дозволити для каталогу Common читання і виконання для всіх користувачів.

40. Створити в каталозі Common файл chat і встановити для нього права на читання й запис для всіх користувачів.

41. Створити (F4) у файлі сценарію ініціалізації оболонки (.bashrc) такі синоніми (alias):

а) «h» – перейти в мій початковий каталог;

б) «dir» – аналог команди dir у Windows NT;

в) «md» – аналог команди md у Windows NT;

г) «rd» – аналог команди rd у Windows NT;

д) «del» – видаляти зазначені файли в «кошик» (каталог Recycled у початковому каталозі);

е) «dird» – очистити «кошик» (каталог Recycled у початковому каталозі);

ж) «bak» – створити резервну копію зазначеного файлу (жорсткий зв'язок у каталозі Bak).

42. Додати в змінну оточення оболонки PATH шлях до каталогу Script свого початкового каталогу.

43. Створити в каталозі Script сценарії оболонки для виконання:

а) переміщення всіх файлів з розширенням «bak» (з чотирма останніми символами імені .bak) зі всіх каталогів свого початкового каталогу в каталог Vak;

б) архівації заданих каталогів з переміщенням архівних файлів у каталог Vak.

44. Призначити створеним файлу сценаріїв оболонки право на виконання і перевірити їх працездатність.

45. Завершити (F10) роботу з Midnight Commander і сеанс роботи з операційною системою.

### **Зміст звіту**

1. Назва та мета лабораторної роботи.

2. Результати, отримані під час виконання лабораторної роботи, та необхідні пояснення, як зазначено у відповідних пунктах робочого завдання.

3. Письмові відповіді на контрольні питання.

### **Контрольні питання**

1. Типи командних оболонок Linux.

2. Змінні оточення, підтримувані оболонкою bash.

3. Призначення та команди файлів .bash\_profile і .bashrc.

**Література:** [1, с. 270–281, 2, с. 168–179, 3, с. 147–173, 5, с. 56–80, 8, с. 166–250].

## **2 КРИТЕРІЇ ОЦІНЮВАННЯ ЯКОСТІ ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ СТУДЕНТАМИ**

У першому семестрі студенти виконують 11 лабораторних робіт з навчальної дисципліни «Обчислювальна техніка».

Загальна кількість балів, яку отримують студенти за виконання лабораторних робіт, максимально складає 30 балів – сума за захист виконаних лабораторних робіт (максимально 20 балів за всі захищені лабораторні роботи, тобто по 1,8 бала на кожну лабораторну роботу) і наявність друкованих звітів про виконання лабораторних робіт (10 балів за наявність всіх одинадцяти звітів, належно оформлених).

Заздалегідь, згідно з розкладом занять, перед виконанням уже безпосередньо експериментальної частини лабораторної роботи студенти повинні підготуватися до її виконання. Підготовка до лабораторної роботи студент проводить самостійно в позааудиторний час.

Підготовка складається із вивчення коротких теоретичних відомостей, що наведені як безпосередньо в цих методичних вказівках, так і в літературі, зазначеній у відповідному розділі кожної лабораторної роботи. Під час підготовки до лабораторної роботи студент також повинен розпочати складання звіту.

Допуск студента до виконання експериментальної частини лабораторної роботи здійснює викладач. Студент повинен подати звіт про підготовку до виконання лабораторної роботи, який містить назву й мету роботи та відповіді на контрольні питання. Інакше студент вважається не готовим до лабораторної роботи, не допускається до виконання її експериментальної частини і продовжує підготовку безпосередньо в лабораторії під керівництвом викладача.

За результатами виконання експериментальної частини студент складає звіт, зміст якого зазначений у відповідному розділі кожної лабораторної роботи. Залік з лабораторної роботи студент одержує після співбесіди з викладачем, у ході якої він повинен продемонструвати теоретичні знання з

теми даної лабораторної роботи і вміти надати вичерпні пояснення за змістом звіту про виконання лабораторної роботи.

### Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		Для екзамену, курсового проекту (роботи), практики	Для заліку
90–100	A	Відмінно	Зараховано
82–89	B	Добре	
74–81	C		
64–73	D	Задовільно	
60–63	E		
35–59	FX	Незадовільно з можливістю повторного складання	Не зараховано з можливістю повторного складання
0–34	F	Незадовільно з обов'язковим повторним вивченням навчальної дисципліни	Не зараховано з обов'язковим повторним вивченням навчальної дисципліни

## СПИСОК ЛІТЕРАТУРИ

1. Колисниченко Д. Н. Linux: полное руководство / Д. Н. Колисниченко, Питер В. Аллен. – С.Пб. : Наука и техника, 2006. – 784 с.
2. Колисниченко Д. Н. Самоучитель Linux. Установка, настройка, использование / Д. Н. Колисниченко. – СПб. : Наука и техника, 2006. – 688 с.
3. Костромин В. А. Самоучитель Linux для пользователя / В. А. Костромин. – СПб. : БХВ-Петербург, 2003. – 672 с.
4. Микляев А. П. Настольная книга пользователя IBM PC / А. П. Микляев. – М. : СОЛОН-Пресс, 1997. – 413 с.
5. Робачевский А. ОС Unix / А. Робачевский. – С.Пб. : БХВ-Петербург, 2002. – 528 с.
6. Скловская С. Л. Команды Linux. Справочник / С. Л. Скловская. – С.Пб. : ООО «ДиаСофтЮП», 2004. – 848 с.
7. Станек У. Командная строка Microsoft Windows. Справочник администратора / У. Станек. – М. : Издательско-торговый дом «Русская редакция», 2004. – 480 с.
8. Тейнсли Д. Linux и UNIX: программирование в shell. Руководство разработчика / Д. Тейнсли. – К. : Издательская группа BHV, 2001. – 464 с.
9. Фигурнов В. Э. IBM PC для пользователя / В. Э. Фигурнов. – М. : Инфра-М, 2002. – 640 с.

Методичні вказівки щодо виконання лабораторних робіт з навчальної дисципліни «Обчислювальна техніка» для студентів денної форми навчання за напрямом 123 – «Комп'ютерна інженерія». Частина II

Укладачі: старш. викл. Ю. В. Зілінський,  
к. т. н., доц. О. Г. Славко

Відповідальний за випуск зав. кафедри КІС А. В. Луговой

Підп. до др. \_\_\_\_\_. Формат 60×84 1/16. Папір тип. Друк ризографія.  
Ум. друк. арк. \_\_\_\_\_. Наклад \_\_\_\_\_ прим. Зам. № \_\_\_\_\_. Безкоштовно.

Видавничий відділ  
Кременчуцького національного університету  
імені Михайла Остроградського  
вул. Першотравнева, 20, м. Кременчук, 39600