***Лабораторна робота № 4***

***«Програмування командних BAT-файлів, обробка аргументів, мітки,***

***оператори переходу і розгалуження, організація циклів»***

***(Термін виконання – 2 тижні!)***

**Увага!** Теоретичний матеріал, викладений у даній лабораторній роботі, а також у наступних, зовсім не претендує на повноту. Даний матеріал використовується у якості методичних вказівок для виконання завдань до ЛР.

Матеріал у повному обсязі представлений у конспекті лекцій та у основній літературі до дисципліни.

***Пакетні командні файли (ПКФ) в системі MS-DOS***

Пакетні командні файли використовуються в операційних системах для того, щоб мати можливість задавати послідовність команд ОС, управляти запуском завдань, створювати різні варіанти робочого середовища.

ПКФ в системі MS-DOS є текстовими файлами в коді ASCII, які містять послідовності команд ОС. Вони повинні мати розширення ***.bat*** і мати ім'я не більше восьми символів. При цьому, в поточному каталозі не повинно бути однойменних файлів з розширенням ***.com*** або ***.ехе***.

В ОС MS-DOS спеціально для ПКФ передбачено вісім команд і спеціальний символ ***@*** (табл.1).

Дані команди є внутрішніми командами ОС, розміщуються в командному процесорі і доступні з будь-якого каталогу.

Табл. 1

|  |  |
| --- | --- |
| **Команда** | **Призначення** |
| ***@*** | Не виводить на екран дисплея рядок, який слідує за ним |
| ***call*** | Здійснює виконання одного пакета команд всередині іншого пакета |
| ***cls*** | Очищення екрану |
| ***echo off*** | Забороняє виведення на екран командних рядків |
| ***echo on*** | Дозволяє виведення на екран командних рядків |
| ***echo <текст>*** | Виведення на екран текстової інформації (при дії команди ***echo off***) |
| ***for*** | Завдання циклу |
| ***goto <мітка>*** | Передача управління на вказану мітку |
| ***if*** | Умовне виконання команди |
| ***pause*** | Переривання виконання програми |
| ***rem*** | Допоміжна команда у пакетних файлах для написання коментарів |
| ***shift*** | Зрушення вліво параметрів команди |

Для запуску ПКФ вводиться його ім'я без розширення. У разі, якщо ПКФ знаходиться не в поточному каталозі, шлях доступу до ПКФ повинен бути визначений командою ***path***. Якщо в якій-небудь команді пакета була допущена помилка, то виконання пакету припиняється і на екран виводиться повідомлення про помилку.

Можна перервати виконання пакету натисканням **Ctrl-C**. У цьому випадку виводиться повідомлення:

***Terminate batch job (Y / N)?***

***(Завершити роботу командного файлу (Так / Ні)?***

ПКФ можна сформувати за допомогою будь-якого текстового редактора або інших програм, що дозволяють створювати файли в коді ASCII.

***Команда ECHO [ON/OFF]***

***ECHO <текст повідомлення>*** - вкл/викл режиму виведення команди перед її виконанням.

За замовчуванням команди пакетного файлу перед виконанням виводяться на екран, що виглядає не дуже естетично. За допомогою команди ***ECHO OFF*** можна відключити дублювання команд, що йдуть після неї (сама команда ***ECHO OFF*** при цьому все ж дублюється). Наприклад,

***REM Наступні дві команди будуть дублюватися на екрані ...***

***DIR C:\***

***ECHO OFF***

***REM А решта вже не будуть***

***DIR D:\***

Для відновлення режиму дублювання використовується команда ***ECHO ON***. Крім цього, можна відключити дублювання будь-якої окремої рядки в командному файлі, написавши на початку цього рядка символ ***@***, наприклад:

***ECHO ON***

***REM Команда DIR C:\ дублюється на екрані***

***DIR C:\***

***REM А команда DIR D:\ - ні***

***@DIR D:\***

Таким чином, якщо поставити в самий початок файлу команду ***@ECHO OFF***, то це вирішить всі проблеми з дублюванням команд.

***@ECHO OFF***

***ECHO Привіт!***

***ECHO.***

***ECHO Бувай!***

При цьому команда ***ECHO.*** (Точка повинна слідувати безпосередньо за словом "ECHO") виводить на екран порожній рядок.

Для виведення службових слів командою ***echo*** дані слова слід помістити між двома символами ***^***, що призначений для екранування.

***Використання параметрів командного рядка***

При запуску пакетних файлів в командному рядку можна вказувати довільне число параметрів, значення яких можна використовувати всередині файлу. Це дозволяє, наприклад, застосовувати один і той же командний файл для виконання команд з різними параметрами.

Для доступу з командного файлу до параметрів командного рядка застосовуються символи ***%0***, ***%1***, ..., ***%9*** або ***%\****. При цьому замість ***%0*** підставляється ім'я виконуваного пакетного файлу, замість ***%1***, ***%2***, ..., ***%9*** - значення перших дев'яти параметрів командного рядка відповідно, а замість ***%\**** - всі аргументи. Якщо в командному рядку при виклику пакетного файлу задано менше дев'яти параметрів, то "зайві" змінні з ***%1*** - ***%9*** заміщуються порожніми рядками. Розглянемо наступний приклад. Нехай є командний файл ***copier.bat*** такого змісту:

***@ECHO OFF***

***CLS***

***ECHO Файл %0 копіює каталог %1 в %2***

***XCOPY %1 %2 /S***

При необхідності можна використовувати більш дев'яти параметрів командного рядка. Це досягається за допомогою команди ***SHIFT***, яка змінює значення заміщаються параметрів з ***%0*** по ***%9***, копіюючи кожен параметр в попередній, тобто значення ***%1*** копіюється в ***%0***, значення ***%2*** – в ***%1*** і т.д. Замість параметру ***%9*** присвоюється значення параметра, наступного в командному рядку за старим значенням ***%9***. Якщо ж такий параметр не заданий, то нове значення ***%9*** - порожній рядок.

***Команда GOTO***

Після імені цієї команди повідомляється мітка командного рядка, до якої виконується перехід. Якщо перед рядком зазначена двокрапка, то це означає, що за ним слідує мітка, а не команда. Між двокрапкою і ім'ям проміжок не ставиться, між ім'ям і командою ***goto*** залишають проміжок. Довжина мітки обмежується 8 символами.

Наприклад,

***goto exit оператор безумовного переходу за міткою***

***…………………………***

***:exit***

Командний файл може містити мітки і команди ***GOTO*** переходу до цими мітками. Будь-який рядок, що починається з двокрапки, сприймається при обробці командного файлу як мітка. Ім'я мітки задається набором символів, що настають за двокрапкою до першого пробілу або кінця рядка. Наведемо приклад.

Нехай є командний файл такого змісту:

***@ECHO OFF***

***XCOPY file1.txt dir1***

***GOTO Label1***

***ECHO Цей рядок ніколи не виконається***

***:Label1***

***REM Продовження виконання***

***DIR dir1***

Після того, як в цьому файлі ми доходимо до команди ***GOTO Label1***, його виконання продовжується з рядка ***REM Продовження виконання***.

У команді переходу всередині файлу ***GOTO*** можна задавати в якості мітки переходу рядок ***:EOF***, яка передає управління в кінець поточного пакетного файлу (це дозволяє легко вийти з пакетного файлу без визначення будь-яких позначок в самому його кінці).

***Команда CALL***

Також для переходу до мітки всередині поточного командного файлу крім команди ***GOTO*** можна використовувати і розглянуту вище команду ***CALL***:

***CALL: мітка\_аргументу***

При виклику такої команди створюється новий контекст поточного пакетного файлу з заданими аргументами, і управління передається на інструкцію, розташовану відразу після мітки. Для виходу з такого пакетного файлу необхідно два рази досягти його кінця. Перший вихід повертає управління на інструкцію, розташовану відразу після рядка ***CALL***, а другий вихід завершує виконання пакетного файлу. Наприклад, якщо запустити з параметром ***"Копія-1"*** командний файл такого змісту:

***@ECHO OFF***

***ECHO %1***

***CALL :2 Копія-2***

***:2***

***ECHO %1***

то на екран виведуться три рядки:

***Копія-1***

***Копія-2***

***Копія-1***

Таким чином, подібне використання команди ***CALL*** дуже схоже на звичайний виклик підпрограм (процедур) в алгоритмічних мовах програмування.

***Команда CHOICE***

Дана команда є єдиною командою, що взаємодіє з кінцевим користувачем у пакетних командних файлах. За допомогою команди ***CHOICE*** користувач може ввести певний один символ для відповіді.

Синтаксис даної команди є наступним:

***choice [/с:список символів] [/t:символ, число секунд] /m:повідомлення***, де:

* ***/c:список символів*** - вказує допустимі символи, які може ввести користувач у відповідь на повідомлення. Якщо даний параметр не вказано, допустимі символи - ***y*** і ***n***.
* ***/t:символ, число секунд*** - якщо цей параметр вказаний, то в разі, коли користувач автоматично через певний час не натиснув на жодну клавішу, приймається відповідь ***символ***.

Наприклад:

***choice /c:yn /m:"запустити програму alfa?"***- при відповіді ***n*** значення змінної ***%errorlevel%***  встановлюється рівним 2, а при відповіді ***y***- рівним 1.

***Оператори умови у командних файлах***

За допомогою команди ***IF ... ELSE*** (ключове слово ***ELSE*** може бути відсутнім) в пакетних файлах можна виконувати обробку умов декількох типів. При цьому якщо задана після ***IF*** умова приймає істинне значення, система виконує наступну за умовою команду (або кілька команд, укладених в круглі дужки), в іншому випадку виконується команда (або кілька команд в дужках), наступні за ключовим словом ***ELSE***.

Синтаксис команди ***IF ... ELSE*** наступний:

***IF <умова>***

***(***

***<блок\_команд1>***

***)***

***[ELSE***

***(***

***<блок\_команд2>***

***)]***

Якщо у блоку команд знаходиться тильки одна команда, квадратні дужки можуть бути опущені.

Умови можуть бути наступних типів:

1. **Перевірка значення змінної:**

***[NOT] <змінна1>==<змінна2>*** - тільки для рядків;

***[NOT] <змінна1> <оператор\_порівняння> <змінна2>*** - для рядків та чисел.

При цьому оператори порівняння можуть бути наступними:

* ***EQL =***
* ***NEQ ≠***
* ***LSS <***
* ***LEQ ≤***
* ***GTR >***
* ***GEQ ≥***

При даному типу умови у команді if можна задати додатковий ключ ***/I*** (відразу після назви команди), який не буде враховувати регістр символів.

1. **Перевірка існування елементу (файлу або каталогу):**

***[NOT] EXIST <елемент>***

Умова вважається істинною, якщо зазначений файл або каталог існує. Лапки для імені не потрібні.

1. **Перевірка наявності змінної середовища**

***DEFINED <змінна>***

Тут умова ***DEFINED*** застосовується подібно умові ***EXISTS*** наявності заданого файлу, але приймає в якості аргументу ім'я змінної середовища і повертає істинне значення, якщо ця змінна визначена.

### Перевірка коду завершення попередньої команди

***[NOT] ERRORLEVEL <число>***

Тут умова вважається істинною, якщо остання запущена команда або програма завершилася з кодом повернення, що є рівним або перевищує вказане число.

***Організація циклів у командних файлах***

У командних файлах для організації циклів використовуються кілька різновидів оператора ***FOR***.

Найпростіший варіант синтаксису команди ***FOR*** для командних файлів має такий вигляд:

***FOR %%змінна IN (множина)***

***DO***

***(***

***<блок\_команд>***

***)***

Слід звернути особливу увагу на те, що перед назвою змінної повинні стояти саме два знака відсотка (***%%***), а не один, як у випадку використання команди ***FOR*** безпосередньо з командного рядка.

Параметр ***<множина>*** в команді ***FOR*** може також представляти одну або кілька груп файлів. Наприклад, щоб вивести в файл список всіх файлів з розширеннями ***txt*** і ***prn***, що знаходяться в каталозі ***C:​\TEXT***, без використання команди ***DIR***, можна використовувати командний файл такого змісту:

***@ECHO OFF***

***FOR %%f IN (C:\TEXT\\*.txt C:\TEXT\\*.prn) DO ECHO %%f >> list.txt***

Різновиди циклу ***FOR*** забезпечуються за допомогою деяких його ключів (ключ потрібно вказати відразу за словом ***FOR***):

* ***/D*** – те саме, що і ***FOR*** за замовчуванням, тільки у множині значень вказуються не файли, а каталоги;
* ***/R*** – розширена версія циклу ***FOR*** з ключем ***/D***, яка дозволяє обробити каталоги, вказані у множині значень, рекурсивно;
* ***/L*** – арифметична версія циклу ***FOR***, що має вигляд: ***FOR /L %%<змінна> IN (<початок>, <крок>, <кінець>)***;
* ***/F*** – найбільш потужна версія циклу ***FOR***, яка призначена для обробки вмісту файлів та має вигляд: ***FOR /F ["ключі"] %%<змінна> IN (<набір>)***. Більш детально про дану версію можна прочитати у конспекті лекцій та літературі, що знаходиться у папці дисципліни.

***Завдання***

1. Написати командний файл, який буде копіювати з поточного каталогу всі файли з розширенням txt, крім одного файлу, вказаного в якості другого параметра командного рядка, в каталог, вказаний першим параметром. Якщо ім'я каталогу, в який повинно проводитися копіювання, не задано, то вивести повідомлення про це, якщо буде введений символ Y - продовжити роботу, тобто створити каталог і скопіювати файли, N - перервати виконання файлу.

Якщо пакетний файл запускається взагалі без параметрів, то вивести опис його синтаксису.

2. Створити командний файл, який виводив би вміст каталогу, зазначеного в якості параметра командного рядка, причому користувачеві повинна бути надана можливість вибору за допомогою меню пристрою для виведення: на екран (інформація виводиться по одному екрану), в текстовий файл catalog.txt .

Командний файл повинен обробляти два ключа:

* ***/A*** - сортування виведеної інформації за алфавітом
* ***/D*** - за датою створення.

Якщо пакетний файл запускається взагалі без параметрів, то вивести опис його синтаксису.

3. Створити пакетний файл, який розкладає вміст папки, ім'я якої передається в якості параметра командного файлу, по підпапках відповідно до розширення файлів - всі***.exe*** файли - в папку ***EXE***, ***.txt*** - в ***TXT*** і т.д. Створити окрему папку для файлів без розширення.

***Контрольні питання***

1. Що таке командний файл?

2. Який файл є командним інтерпретатором?

3. Які команди операційної системи можуть використовуватися в **ПКФ**?

4. Як запускаються **ПКФ**?

5. Яким чином можна перервати виконання пакетного командного файлу?

6. Яким чином організовується розгалуження у **ПКФ**?

7. Скільки параметрів можна використовувати при виклику **ПКФ**?

8. Як діє команда ***call***?

9. Яке призначення символу ***@***?