**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Чорноморський національний університет   
імені Петра Могили**

**Факультет комп’ютерних наук**

**Кафедра Інтелектуальних інформаційних систем**

****

**Лабораторна робота №10**

**Варіант №8**

Дисципліна "Операційні системи"

**Виконав:**

**Студент 201 групи**

Грабовський Єгор

\_\_\_\_\_\_\_\_\_\_\_\_\_(підпис)

\_\_\_\_.\_\_\_\_20\_\_р.(дата)

**Викладач**

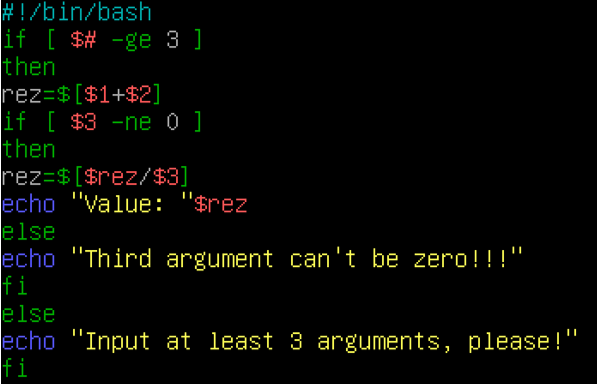
Горбань Г.В.

\_\_\_\_\_\_\_\_\_\_\_\_\_(підпис)

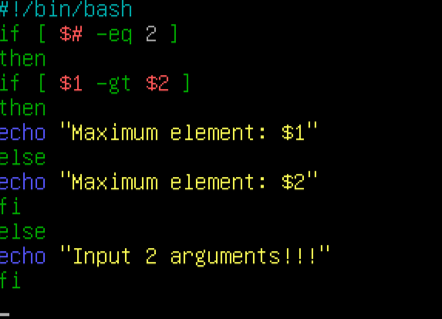
\_\_\_\_.\_\_\_\_20\_\_р.(дата)

**Миколаїв – 2020**

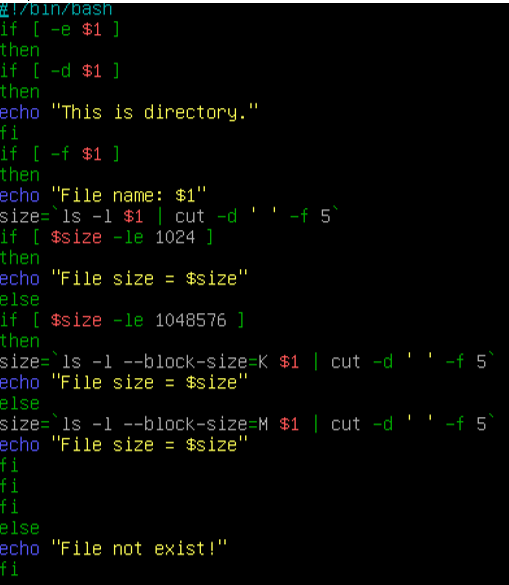
***Завдання 1)***



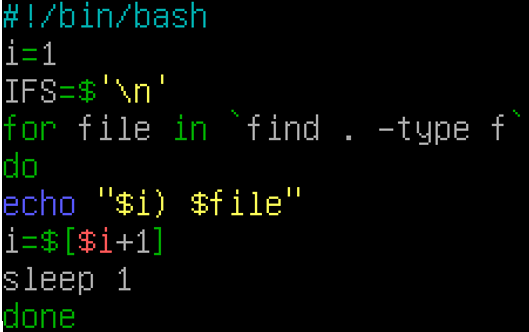
***Завдання 2)***



***Завдання 3)***



***Завдання 4)***



***Завдання 5)***



***Контрольні питання***

1. Яким чином в програму можна передати аргументи?

Довільна кількість аргументів вказується після відповідної команди.

1. Як в програмі можна отримати доступ до аргументів, з якими викликалася програма?

За допомогою спеціальних змінних : $n , де n – номер змінної, або ${n}, якщо номер змінної є числом з більш ніж однієї цифри.

1. Які спеціальні змінні можна використовувати в скриптах командного інтерпретатора?

$0 - ім’я сценарію;

$\* - всі аргументи одним рядком;

$@ - всі аргументи окремо;

$# - кількість аргументів;

$$- ідентифікаційний номер поточного процесу;

$! - ідентифікаційний номер фонового завдання;

$?- ідентифікаційний номер виконуваної команди.

1. Яким чином можна запускати програми в залежності від виконання інших програм?

Існує змінна середовища оточення, яка зберігає статус виконання останньої команди. Оператори if-then дозволяють виконати певні команди в тому випадку, якщо статус виконання вказаної після if команди нульовий.

1. Які оператори розгалудження існують у bash? Наведіть їх синтаксис.

Оператор if-then-else виконує команди після then якщо команда після if має нульовий статус виконання і команди після else в протилежному випадку.

if <команда>

then

<команди>

else

<команди>

fi

Замість команди після if можна вказувати умову в квадратних дужках. Так можна вказувати ті ж умови, які можуть вказуватись для команди test.

Оператор case дозволяє зробити вибір з багатьох альтернатив.

case <змінна> in

<шаблон 1>) <блок команд> ;;

<шаблон 2>) <блок команд> ;;

\*) <блок команд за замовчуванням> ;;

1. Для чого призначена та який синтаксис має команда test?

Команда test перевіряє зазначену умову і закінчує свою роботу з кодом завершення 0 якщо умова виконується і 1 якщо ні. Цією командою можна перевіряти характеристики файлів, порівнювати рядки або числа.

Команда test для перевірки характеристик файлів має наступний синтаксис:

test <опція> <ім’я файлу>

Опція може мати наступні значення:

|  |  |
| --- | --- |
| Опція | Опис |
| -b | Істина, якщо файл існує і є блоковим спеціальним файлом |
| -c | Істина, якщо файл існує і є символьним спеціальним файлом |
| -d | Істина, якщо файл існує і є каталогом |
| -e | Істина, якщо файл існує |
| -f | Істина, якщо файл існує і є звичайним файлом |
| -h | Істина, якщо файл існує і є символічною посиланням |
| -r | Істина, якщо файл існує і доступний на читання |
| -s | Істина, якщо файл існує і має ненульовий розмір |
| -S | Істина, якщо файл існує і є сокетом |
| -w | Істина, якщо файл існує і доступний за попереднім записом |
| -x | Істина, якщо файл існує і доступний на виконання |

Команда test для порівняння рядків може мати наступні опціії

|  |  |
| --- | --- |
| Опція | Опис |
| -z <рядок> | Істина, якщо рядок має нульову довжину |
| -n <рядок> | Істина, якщо рядок має ненульову довжину |
| <рядок1> = <рядок2> | Істина, якщо рядки збігаються |
| <рядок1>! = <рядок2> | Істина, якщо рядки різні |

Команда test для порівняння файлів має такий синтаксис:

test <число1> <оператор> <число2>

Можуть використовуватись наступні оператори:

|  |  |
| --- | --- |
| Оператор | Опис |
| -eq | Істина, якщо числа рівні |
| -ne | Істина, якщо числа не рівні |
| -lt | Істина, якщо менше |
| -le | Істина, якщо менше або дорівнює |
| -gt | Істина, якщо більше |
| -ge | Істина, якщо більше або дорівнюють |

1. Які види циклів існують у bash? Наведіть їх синтаксис.

В bash існують такі види циклів:

* for

Цикл for виконує команди зі списку команд для кожного елемента зі списку і має наступний

синтаксис:

for <ім'я> in <елемент1> <елемент2> ... <елементN>

do

<список команд>

done

* while

Оператор циклу while має наступний синтаксис:

while <команда>

do

<список команд>

done

З кожною ітерацією циклу спочатку виконується команда після слова while. Якщо результат ненульовий, то відбувається вихід з циклу, якщо нульовий – виконуються команди в списку команд і відбувається перехід до наступної ітерації.

* select

Цикл select дозволяє створювати зручні меню. Він корисний, коли необхідно, щоб користувач вибрав один елемент із запропонованого списку. Оператор select має такий же вигляд, як і оператор for, за винятком ключового слова. При виконанні даного оператора циклу всі елементи зі списку висвічуються на екрані разом з їх порядковими номерами, після чого з'являється спеціальне запрошення для введення. Після того як користувач зробив вибір, виконуються команди зі списку команд, після чого виконання циклу починається заново з виведенням запрошення для введення.

Для переривання нормальної роботи циклів існують оператори break і continue, які працюють так само, як їх аналоги в мові C.