**ΗΜΥ 316**

**Εργαστήριο Λειτουργικών Συστημάτων και Δικτύων**

**Πανεπιστήμιο Κύπρου**

**Φοίβος Λύμπουρας 1016477**

**Εντουίνα Κάρουλλα 1042364**

**Στέλιος Καραγιώργης 1021340**

**Θεοδόσιος Ιωάννου 1020844**

**Team 7**

**Assignment 5**

**Due Date 24 Feb**

**Exercise 1**

**Objectives:**

* Create shell script to welcome user.
* The greeting message should be different based on the time.

**Main solution:**

We take the user name, host name and date, then we go through some if statements based on the time of the time of day. The different times display different messages.

**Text

Description automatically generatedText

Description automatically generatedResult examples:**

**Pseudo code:**

If hours between 5 and 12 am

Print good morning user

If hours between 12 and 18 pm

Print good afternoon user

If hours between 18 and 24

Print good evening user

Else print its too late user go to sleep

**Exercise 2**

**Objectives:**

* Get number n
* Make the calculations based on the number given
* Display the result of those calculations

**Main solution:**

We take the number given to us by the user and run it through the according calculations based on the result we want

**Text

Description automatically generatedText

Description automatically generatedResults:**

**Pseudocode:**

Read number n

For I =1 to n

M= (i\*2)

Print factorial

While I less than n/2

If n mod I equals 0

Then f=1

I++

If f equals 1

N is not prime

Else its prime

**Exercise 3**

**Objectives:**

* Create a program with 4 procecess.
* Each process need to print a total of 100 messages
* Each process will print 3 messages at a time
* The processes cycle in the following fashion 3 -> 2 -> 1 -> 4

**Main solution:**

We create 4 processes that will be printing their according messages. Then by using pipes we direct which process will be running after the previous one is put on a hold. We also direct the flow of which the sequence of processes run.

**A picture containing graphical user interface

Description automatically generatedText

Description automatically generatedResults:**

**Pseudocode:**

Pipe the processes

Fork to create child processes

If cases for each coresponding child processes

For I =1 to Max messages

Close write pipe

Read from pipe

For k=1 to number of prints for each processes

Print message and number

If number passes max count

Stop

Close next read pipe

Write in next pipe

**Exercise 4**

**Objectives:**

* Create program where any number of processes and messages can be input

**Main solution:**

The program will run based on the amount of processes we give it and on how many messages to be printed per cycle. So we follow the same procedure from before except now all processes are directed itteratively.

**Text

Description automatically generatedText

Description automatically generatedResults:**

**Pseudocode:**

Pipe the processes

Fork to create child processes

For I =1 to Max messages

Close write pipe

Read from pipe

For k=1 to number of prints for each processes

Print message and number

If number passes max count

Stop

Close next read pipe

Write in next pipe

1. A process knows that it is time to read once the write ( p[1] ) part of the pipe is closed
2. The child processes exit code will be lost, and the child process will become orphaned until it is terminated.