Alexandria University
Faculty of Engineering
Comp. & Comm. Engineering
CC373: Operating Systems

جامعة الاسكندرية
كلية الهندسة
برنامج هندسة الحاسب والاتصالات
مادة نظم التشغيل

Lab1
Simple Shell Commands

Objectives

- To get familiar with Linux and its programming environment.
- To understand the relationship between OS command interpreters (shells), system calls, and the kernel.
- To learn how to write .sh file
- To learn what is processes and how to fork and what does fork mean

Overview

A Unix shell is a command-line interpreter or shell that provides a traditional user interface for the Unix operating system and for Unix-like systems. The shell can runs in an **interactive** mode. In interactive mode, you will display a prompt (e.g. **`Shell>`**) and the user of the shell will type in a command at the prompt (e.g. **`Shell> ls -l`**) Your shell terminates when the user enters the exit command at the prompt.

List of shell commands:

- **`cd`** -- change directory
- **`pwd`** -- print working directory
- **`ls`** -- list all files and sub-directories in a directory
- **`stat`** -- display information about a file
- **`rm`** -- remove (i.e. delete) a file
- **`cp`** -- copy a file
- **`cat`**, **`more`**, **`less`** -- list the contents of a file
- **`chmod`** -- change file mode, i.e. file permissions
- **`ln`** -- create a link
- **`wc`** -- count words
- **`mkdir`** -- create a new directory
- **`head`** -- output only the first lines of a file
- **`tail`** -- output only the last lines of a file
- **`grep`** -- find a word in one or more files
- **`ps`** -- process status (lists running processes, often run as **ps aux** for the most

information)

- **`cut`** -- extract a column from a file
- **`sort`** -- sort a file alphabetically
- **`uniq`** -- remove adjacent duplicate lines
- **`| pipe`** -- the `pipe` command lets you sends the output of one command to another format is `Command-1 | Command-2 | …| Command-N`
- **`>>`** or **`>`** or **`<<`** or **`<`** -- To redirect a standard output of any command to another process, use the `>` symbol. To redirect a standard input of any command, use the `<` symbol.
- **`ls | wc -l >> text.txt`** what do you think this command do ?

Write and run your first .sh file

```
#!/bin/sh
#
# This shell script prints hello to all the friends you
# pass as parameter
#

if [ $# -le 1 ]
then
  echo
  echo "$0 needs at least one argument"
  echo "  Eg."
  echo "     $0 Mickey Donald Daisy"
fi

for friend in $*
do
  echo "Hello $friend"
done
```

run the file using your friends names

Count files in directory Example

```
#!/bin/bash
#
# Counts how many files are in the directories passed
# as parameter. If not directories are passed it uses
# the current directory.

# If no arguments use only current directory
if [ $# -lt 1 ]
then
  dirs=.
else
  dirs=$*
fi

#Initialize file counter to 0
count=0

# for all the directories passed as argument
for dir in $dirs
do
  echo $dir:
  for file in $dir/*
  do
     echo "$count: $file"
     count=`expr $count + 1`
  done
done

echo "$count files found"
```

Try to run the code by passing no argument then passing some folder names

Policies

- Marks is on attending the lab and running the code over there.