Exercise Sheet 2

Exercise 1 (Time-Sharing)

1. Describe the objective of time-sharing.

2. Describe how time-sharing distributes the computing time among the processes.

3. Give the name of the pseudo parallel program or process execution.

4. Describe the objective of the pseudo parallel program or process execution.

5. Describe what scheduling is.

6. Describe what swapping is.

7. Describe how memory protection works.

8. Describe the purpose of memory protection.

Exercise 2 (Basic Linux/UNIX commands)

Which command is used to...

1. check the man pages?

2. print out the present working directory in the shell?

3. create a new directory?

4. navigate to a directory?

5. print out the content a directory in the shell?

6. create an empty file?

7. try to determine the content of a file?

8. concatenate the content of files with other files and can also be used to print out the content of a file?

9. print out lines from the end of a file in the shell?

10. print out lines from the beginning of a file in the shell?

11. copy files or directories to a different location?

12. move files or directories to a different location?

13. delete files or directories?

14. delete an empty directory?

15. output a string in the shell?

16. modify the permissions of the file or directory?

17. change the password of a user?

18. terminate a session (and thus shell) and allows to specify the return value of she shell script?

19. reboot the system?

20. shut the system down?

21. create a new user?

22. delete a user?

23. modify a user?

24. print out the group memberships of a user?

25. create a new group?

26. delete a group?

27. change a group?

28. change the user ($\Longrightarrow$ ownership) which is associated with a file or directory?

29. change the group which is associated with a file or directory?

30. create a link?

31. search a file for lines which contain a search pattern?

32. print out a list of running processes in the shell?

33. bring a process, running in the backgrund of the shell, into foreground?

34. bring a process into the background of the shell?

35. kill (terminate) a process?

36. kill (terminate) a group of processes?

37. specify the priority of a new process?

38. modify the priority of an existing process?

39. print out the process tree in the shell?

Exercise 3 (System Programming)

This small programming task is designed to refresh your knowledge about programming in `C` and familiarize yourself with system programming on a *nix system.

In this assignment, you will write your own version of the `ls` command which lists files in a directory. Directories are actually special files that contain information about other files. Your version of ls, **listdir** will be significantly simpler than the standard ls.

You should provide a Makefile to build your application.

As an editor you can use **micro** (`https://micro-editor.github.io/`).

Basic functionality when user executes **listdir**:

1. If no command line arguments are given, the current directory is listed with a newline after each file including the last one. By default, any file starting whose first character is `.` will not be listed unless the hidden flag is set (see below).

2. An optional flag, `-h` (for hidden), may be specified which will list all files or directories including those that start with a '`.`'.

3. One or more command line arguments may be given. Each argument is a directory to list, i.e., `listdir a02 a03` should list the files in directories `a02` and `a03`. If any of the specified directory files cannot be accessed, the error `Cannot access a02` should be printed.

4. You must create a *Makefile* such that when someone types `make` in your working directory it will compile the program with an output of `listdir`.

You will need the system commands `opendir`, `closedir`, and `readdir` to complete this task. For details on how to use these, you can use UNIX's man pages. There is also an online version at `https://www.kernel.org/doc/man-pages/`.

You may find it easier to use `getopt` to parse the -h flag, but this is not required.