# Advanced OS Assignments

## Write a following program in 'C

1. To create 'n' children. When the children will terminate, display total cumulative time children spent in user and kernel mode.
2. To generate parent process to write unnamed pipe and will read from it.
3. To create a file with hole in it.
4. Takes multiple files as Command Line Arguments and print their inode number.
5. To handle the two-way communication between parent and child using pipe.
6. Print the type of file where file name accepted through Command Line.
7. To demonstrate the use of atexit() function.
8. Open a file goes to sleep for 15 seconds before terminating.
9. To print the size of the file.
10. Read the current directory and display the name of the files, no of files in current directory.
11. Write a C program to implement the following unix/linux command (use fork, pipe and exec system call)
    ls –l | wc –l
12. Write a C program to display all the files from current directory which are created in particular month
13. Write a C program to display all the files from current directory whose size is greater that n Bytes Where n is accept from user.
14. Write a C program to implement the following unix/linux command
    i. ls –l > output.txt
15. Write a C program which display the information of a given file similar to given by the unix / linux command ls –l <file name>
16. Write a C program that behaves like a shell (command interpreter). It has its own prompt say "NewShell$". Any normal shell command is executed from your shell by starting a child process to execute the system program corresponding to the command. It should additionally interpret the following command.
    i) count c <filename> - print number of characters in file
    ii) count w <filename> - print number of words in file
    iii) count l <filename> - print number of lines in file
17. Write a C program that behaves like a shell (command interpreter). It has its own prompt say "NewShell$". Any normal shell command is executed from your shell by starting a child process to execute the system program corresponding to the command. It should additionally interpret the following command.
    i) list f <dirname> - print name of all files in directory
    ii) list n <dirname> - print number of all entries
    iii) list i<dirname> - print name and inode of all files

18. Write a C program that behaves like a shell (command interpreter). It has its own prompt say “NewShell$”. Any normal shell command is executed from your shell by starting a child process to execute the system program corresponding to the command. It should additionally interpret the following command.
    i) typeline +10 <filename> - print first 10 lines of file
    ii) typeline -20 <filename> - print last 20 lines of file
    iii) typeline a <filename> - print all lines of file

19. Write a C program that behaves like a shell (command interpreter). It has its own prompt say “NewShell$”.Any normal shell command is executed from your shell by starting a child process to execute the system program corresponding to the command. It should
    i) additionally interpret the following command.
    ii) search f <pattern> <filename> - search first occurrence of pattern in filename
    iii) search c <pattern> <filename> - count no. of occurrences of pattern in filename
    iv) search a <pattern> <filename> - search all occurrences of pattern in filename

20. Write a C program which receives file names as command line arguments and display those filenames in ascending order according to their sizes.
    i) (e.g $ a.out a.txt b.txt c.txt, …)

21. Write a C program which create a child process which catch a signal sighup, sigint and sigquit. The Parent process send a sighup or sigint signal after every 3 seconds, at the end of 30 second parent send sigquit signal to child and child terminates my displaying message "My DADDY has Killed me!!!”.

22. Write a C program to implement the following unix/linux command (use fork, pipe and exec system call). Your program should block the signal Ctrl-C and Ctrl-\ signal during the execution.
    i. ls –l | wc –l
23. Write a C Program that demonstrates redirection of standard output to a file.
24. Write a program that illustrates how to execute two commands concurrently with a pipe.
25. Write a C program that illustrates suspending and resuming processes using signals.
26. Write a C program that illustrates inters process communication using shared memory.