SSN COLLEGE OF ENGINEERING, KALAVAKKAM
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
CS8461 - OPERATING SYSTEM LAB

Lab Exercise 1 Study of System calls and System Commands

1. Study the following system calls and system commands (using Linux manual pages) and write a description and an example for each of the commands in your observation.

System Calls:
Write System call name, syntax, description, header files, input arguments, return value

1. Fork
2. Exec
3. Getpid
4. Getppid
5. Exit
6. Wait
7. Close
8. Opendir
9. Readdir
10. Open
11. Read
12. Write
13. Creat
14. Sleep
15. Pipe

System Commands: Write command name, syntax, description, options and use

1. Cp – i, f, r, v
2. Mv – i, f, n, v
3. Ls – a, c, i, l, 1, r, R
4. Grep – v, c, n, L
5. Chmod – R, v, f, as octal number, u+r, g-w
6. Cat – simple, display n files, > , >>, <
7. Mkdir – v, h, m
8. Rm – f, i, r, R, d, v
9. Rmdir – p, v
10. Wc – c, m, l, w
11. Who – b, p, u, l, d
12. pipe (pipe symbol) - |
13. head – c, n, v, z, q
14. tail – c, n, v
15. nl – I, n, v, s, b
16. awk - f, v

```
1.  main()
    {
            fork();
            printf(“Hello world\n”);
    }
```

```
2.  main()
    {
            printf(“before fork\n”);
            fork();
            printf(“Hello world\n”);
    }
```

```
3.  main()
    {
            pid_t  pro_id;
            printf(“before fork\n”);
            pro_id=fork();
            if(pro_id==0) // Child process
            {
                    printf(“Hello world from child\n”);
            }
            else if(pro_id > 0) // parent process
            {
                    printf(“Hello world from parent\n”);
            }
            else
                    printf(“fork failure\n”)

    }
```

```
4.  main()
    {
            pid_t  pro_id;
            printf(“before fork\n”);
            pro_id=fork();
            if(pro_id==0) // Child process
            {
                    for(i=0, i<10;i++)
                            printf(“Hello world from child %d\n”,i);
            }
            else if(pro_id > 0) // parent process
    {
        for(i=0, i<10;i++)
            printf(“Hello world from parent %d\n”, i);
    }
    else
        printf(“fork failure\n”)

}
```

5. getpid(), getppid()
6. sleep(), wait(), exit()
7. main()

```
{
    pid_t  pro_id;
    printf(“before fork\n”);
    pro_id=fork();
    if(pro_id==0) // Child process
    {
        execl(“/bin/ls”,”ls”,NULL)
        printf(“Hello world from child\n”);
    }
    else if(pro_id > 0) // parent process
    {
        printf(“Hello world from parent\n”);
    }
    else
        printf(“fork failure\n”)

}
```

8. stat, opendir, readdir, open, read, write, close