***Learning Objectives:*** *Students will be able to learn UNIX process management system calls and library functions.*

**Exercise 1**

Write a C program to print the process ID of the process and it’s parent process ID.

**fork ( ) System call**

**Exercise 2**

#include <stdio.h>

main()

{

printf("I am Parent\n");

fork();

printf("Hello World...!\n");

}

**Exercise 3**

#include <stdio.h>

main()

{

int ret;

printf("I am Parent\n");

ret = fork();

printf("Return Value: %d\n", ret);

}

**getpid ( ) and getppid ( ) system calls**

**Exercise 5**

#include <stdio.h>

main()

{

int ret;

printf("Hello World\n");

ret = fork();

if(ret == 0){

printf("I am Child and Return Value=%d\n", ret); printf("Child PID: %d\n", getpid());

printf("Child's Parent PID: %d\n", getppid());

}

else{

printf("I am Parent and Return Value=%d\n", ret);

printf("Parent PID: %d\n", getpid());

}

sleep(20);

}

**execl ( ) system call**

**Exercise 6**

#include <stdio.h>

main()

{

printf(“Here comes the date. \n”);

execl(“/bin/date”, “date”, 0); /\*0 means end-of-arguments \*/

printf(“That was the date. \n”);

}

**Exercise 7**

#include <stdio.h>

main()

{

printf(“Here comes the date. \n”);

fork();

execl(“/bin/date”, “date”, 0);

printf(“That was the date. \n”);

}

Why did you get date two times? and Why didn’t you get first print statement two times?

…………………………………………………………………………………………………

**Exercise 8**

**system ( ) library function**

#include <stdio.h>

main()

{

printf(“Here comes the date. \n”);

system(“date”);

printf(“That was the date”);

}

**Zombi Process**

**Exercise 10**

#include <stdio.h>

main()

{

int id;

if ((id = fork())== 0)

{

printf(“I am child process \n”);

}

else

{

while(1)

sleep(100);

}

}

**Orphan Process**

**Exercise 11**

#include <stdio.h>

main()

{

int id;

if ((id = fork())== 0)

{

printf(“I am child process \n”);

sleep(10);

}

else

{

printf(“I am parent process \n”);

}

}