Task 3

a.Illustrate System calls of UNIX operating system (fork, exec, getpid, exit, wait)

Program:

```
#include<sys/types.h>
 #include<stdio.h>
 #include<unistd.h>
 int main()
 { int pid; pid=fork();
if(pid>0)
printf("I am Parent");
else
 printf("I am child");
 return 0;
}
```

b. User want to replaces the executable file of running process to another executable file using the system calls - EXECL, LS, WAIT.

Program:

```
#include<sys/types.h>
 #include<stdio.h>
 #include<unistd.h>
int main(int argc, char *argv[]) {
 printf("PID of example.c = %d\n", getpid());
char *args[] = {"Hello", "C", "Programming", NULL};
execv("./hello", args);
printf("Back to example.c");
return 0;
}
```

hello.c:

```
#include<sys/types.h>
 #include<stdio.h>
 #include<unistd.h>
int main(int argc, char *argv[]) {
printf("We are in Hello.c\n");
 printf("PID of hello.c = %d\n", getpid());
 return 0;
}
```

Program: ls and wait

```
#include<stdio.h>
 #include<unistd.h>
#include<sys/types.h>
#include<sys/wait.>
#include<stdlib.h>
int main() {
 int pid;
pid = fork();
 if ( pid < 0 )
 {
 printf("\nFork failed\n"); exit (-1);
} else if ( pid == 0 )
 }
 execlp ( "/bin/ls", "ls", "-l", NULL );
 }
 else {
 wait (NULL);
  printf("\nchild complete\n");
exit (0);
 }
 return 0;
}
```