CSE2005- Operating Systems Embedded Lab

Lab Ex. 3 System calls related to Process

1. Write a C program to perform the following tasks:

- Parent process gets the register number from the user(Eg., 20bce1234), forks a child process and sends the register number to the child process.

- Child process computes the department that the regno belongs to and displays the output (Department Name like SCOPE or SENSE or SELECT or …).

- Parent waits until the child terminates and resumes its operation.

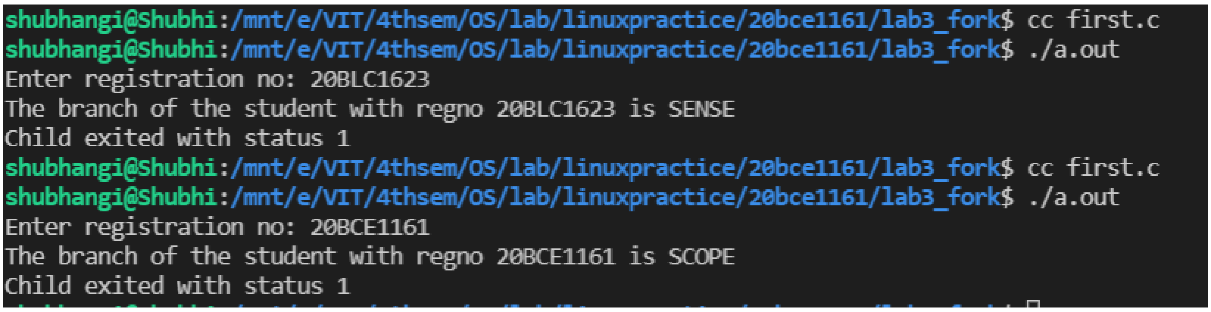
- Finally Parent process terminates after printing the status message from the terminated child

CODE:

```
 /*1.    Write a C program to perform the following tasks:


•   Parent process gets the register number from the user(Eg., 20bce1234), forks a child process and sends the register number to the child process.

•   Child process computes the department that the regno belongs to and displays the output (Department Name like SCOPE or SENSE or SELECT or …).

•   Parent waits until the child terminates and resumes its operation.

•   Finally Parent process terminates after printing the status message from the terminated child

*/

#include<stdio.h>

#include<unistd.h>

#include<sys/wait.h>
```

```c
#include<string.h>
int main(){
    int pipe1[2],ret,rs;
    int status;
    char x[3];
    pid_t pid,ppid;
    char regno[10],rec_regno[10];
    printf("Enter registration no: ");
    rs=pipe(pipe1);
    scanf("%s",regno);
    char *code[10]={"BCE","BLC","BEC"};
    char branch[10][10]={"SCOPE","SENSE","SELECT"};
    pid=fork();
    if(pid!=0){
        write(pipe1[1],regno,sizeof(regno));
        ppid=wait(&status);
        if(WIFEXITED(status)){
            printf("Child exited with status %d\n",WIFEXITED(status));
        }
    }
    else{
        read(pipe1[0],rec_regno,sizeof(rec_regno));
        int j=0;
        for(int i=2;i<5;i++,j++){
        x[j]=rec_regno[i];
        }

        x[j]='\0';

        for(int i=0;i<3;i++){

            if(strcmp(x,code[i])==0){


                printf("The branch of the student with regno %s is %s\n",rec_regno, branch[i]);

                break;

            }

        }





    }


}
```

OUTPUT:



```
shubhangi@Shubhi:/mnt/e/VIT/4thsem/OS/lab/linuxpractice/20bce1161/lab3_fork$ cc first.c
shubhangi@Shubhi:/mnt/e/VIT/4thsem/OS/lab/linuxpractice/20bce1161/lab3_fork$ ./a.out
Enter registration no: 20BLC1623
The branch of the student with regno 20BLC1623 is SENSE
Child exited with status 1
shubhangi@Shubhi:/mnt/e/VIT/4thsem/OS/lab/linuxpractice/20bce1161/lab3_fork$ cc first.c
shubhangi@Shubhi:/mnt/e/VIT/4thsem/OS/lab/linuxpractice/20bce1161/lab3_fork$ ./a.out
Enter registration no: 20BCE1161
The branch of the student with regno 20BCE1161 is SCOPE
Child exited with status 1
```

2. Write a C program to perform the following tasks:

- Parent process forks a child process
- Child process executes a separate program which computes and displays the sum of integers in the given register number.
- Parent waits until the child terminates and resumes its operation.
- Finally Parent process terminates after printing the status message from the terminated child

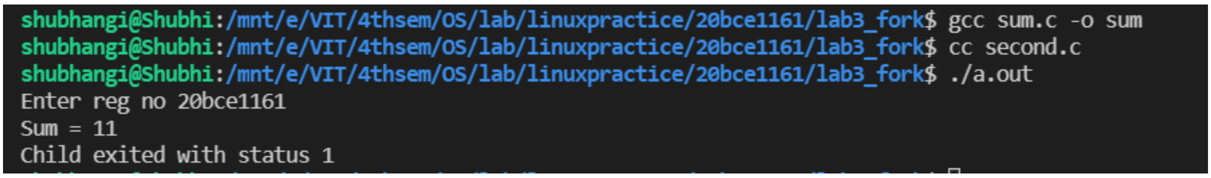
CODE:

```
/*2.    Write a C program to perform the following tasks:

•    Parent process  forks a child process

•    Child process executes a separate program which computes and
displays the sum of integers  in the given register number.

•    Parent waits until the child terminates and resumes its operation.

•    Finally Parent process terminates after printing the status message
from the terminated child

*/


#include<stdio.h>

#include<unistd.h>

#include<sys/wait.h>

#include<string.h>

int main(){

    int status;

    pid_t pid,ppid;

    char regno[10],rec_regno[10];

    pid=fork();

    if(pid!=0){
        ppid=wait(&status);
        if(WIFEXITED(status)){
            printf("\nChild exited with status
%d\n",WIFEXITED(status));
        }
    }
    else{
        char *args[]={"./sum",NULL};
        execvp(args[0],args);


    }


}
```

Sum.c :

```
#include<stdio.h>
#include<stdlib.h>
int main(){
    char reg[10];
    printf("Enter reg no ");
    scanf("%s",reg);
    int ans=0;
    for(int i=0;i<9;i++){
        if(i==2||i==3||i==4)
        continue;
        int x=reg[i]-'0';

        ans+=x;

    }

    printf("Sum = %d",ans);









}
```

Output:



```
shubhangi@Shubhi:/mnt/e/VIT/4thsem/OS/lab/linuxpractice/20bce1161/lab3_fork$ gcc sum.c -o sum
shubhangi@Shubhi:/mnt/e/VIT/4thsem/OS/lab/linuxpractice/20bce1161/lab3_fork$ cc second.c
shubhangi@Shubhi:/mnt/e/VIT/4thsem/OS/lab/linuxpractice/20bce1161/lab3_fork$ ./a.out
Enter reg no 20bce1161
Sum = 11
Child exited with status 1
```