Lab Exercise 2

Process Forking

Name: Abhinav Pandey Roll No. : AM.EN.U4AIE21088

1. What is the output of the following code that you are getting?

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>

int main ()
{
        int id, ret;

        ret = fork();
        id = getpid();
        printf("\n My identifier is ID = [%d]\n", id);
```

Answer :

```c
// Abhinav Pandey
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>

int main() {
    int id, ret;

    ret = fork();
    id = getpid();
    printf("\n My id = [%d]\n", id);
    return 0;
}
```

```
My id = [1247]
My id = [1248]
```

2. What is the output of the following code? How many processes are being created including the parent process? Draw the process graph to trace the fork calls.

Answer:

```c
// Abhinav Pandey
#include <stdio.h>

int main() {
    int id, ret;

    ret = fork();
    ret = fork();

    id = getpid();
    printf("\n My id = [%d]\n", id);
    return 0;
}
```

```
My id = [553]
My id = [552]
My id = [551]
My id = [554]
```

3. What is the output of the following code?

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>

void fork3()
{
        int ret;

        ret = fork();
        if (ret == 0)
                printf("\n [%d] Hello from child", getpid());
        else
                printf("\n [%d] Hello from parent", getpid());
}

int main ()
{
        fork3();
        return 0;
}
```

Answer:

```
1  // Abhinav Pandey
2  #include <stdio.h>
3
4  void fork3() {
5      int ret;
6
7      ret = fork();
8
9      if(ret == 0)
10         printf("\n [%d] Hello from child \n",getpid() );
11     else
12         printf("\n [%d] Hello from parent \n",getpid() );
13
14 }
15
16 int main(){
17     fork3();
18     return 0;
19 }
```

```
/tmp/QyJUpg8ibx.o
[1731] Hello from parent
[1732] Hello from child
```

4. What is the output of the following code? How many processes are being created including the parent process? Draw the process graph to trace the fork calls.

```c
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
void
fork4 ()
{
  printf ("\n[%d] L0 \n", getpid ());
  fork ();
  printf ("\n[%d] L1 \n", getpid ());
  fork ();
  printf ("\n[%d] bye \n", getpid ());
}

int
main ()
{
  fork4 ();
  return 0;
}
```

Answer:

```c
// Abhinav Pandey
#include <stdio.h>

void fork4(){
    printf("\n[%d] L0 \n", getpid());
    fork();
    printf("\n[%d] L1 \n", getpid());
    fork();
    printf("\n[%d] bye \n", getpid());
}

int main(){
    fork4();
    return 0;
}
```

```
[1819] L0
[1819] L1

[1820] L1

[1819] bye

[1820] bye

[1822] bye

[1821] bye
```

5. Write a C program called sumfact.c that does the following:
   1. Takes an integer argument (say, N1) from the command line.

   2. Forks two children processes

   a. First child computes 1+2+...+N1 (sum of positive integers up to N1) and prints out the result and its own process ID.

   b. Second child computes 1*2*...*N1 (the factorial of N1) and prints out the result and its own process ID.

   3. Parent waits until both children are finished, then prints out the message “Done” and its own process ID.

```c
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
# include <stdio.h>

void  sum_of_positive(int a){

    int s = 0;
    for(int i=1;i<=a;i++){
        s=s+i;
    }
    printf("Sum of positive numbers is %d",s);
    printf("\n Identifier ID is [%d] \n ",getpid());
}

void sum_of_factorial(int a){
    int s1 = 1;
    for(int i=1;i<=a;i++){
        s1=s1*i;
    }
    printf("Sum of factorial is %d",s1);
    printf("\n Identifier ID is [%d] \n ",getpid());
}

int main(){
    int n, ret,ret1;
    printf("%s","Enter The Number: ");
    scanf("%d",&n);
    ret = fork();
    if(ret ==0){
        ret1 = fork();
        if(ret1 == 0){
            sum_of_factorial(n);
        }
        else{
            sum_of_positive(n);
        }
    }

    else{
    wait(NULL);
    wait(NULL);
    printf("\n DONE \n");
    printf("\n Identifier ID is [%d] \n ",getpid());

    }

}
```