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

#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);
}

```

```
My identifier is ID =[675]
```

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.

```c
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
int
main ()
{
  int id, ret;
  ret = fork ();
  ret = fork ();
  id = getpid ();
  printf ("\n My identifier is ID =[%d] \n", id);
  wait(NULL);
  return 0;
}
```

Answer:

main.c

```c
//Abhinav Pandey

#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
int main ()
{
int id, ret;
ret = fork ();
ret = fork ();
id = getpid ();
printf ("\n My identifier is ID =[%d] \n", id);
wait (NULL);
return 0;
}

```

```
My identifier is ID =[1193]

My identifier is ID =[1194]
```

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:

```c
//Abhinav Pandey

#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;
}
```

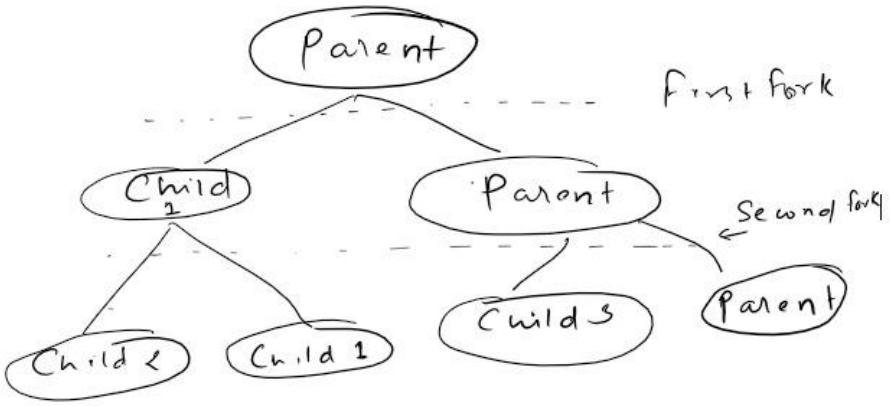
```
[321] Hello from parent
```

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:



main.c

```c
//Abhinav Pandey

#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;
}

```

```
[422] L0

[422] L1

[426] L1

[422] 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.

Answer:

main.c

```c
#include <unistd.h>
#include<stdio.h>
#include<stdlib.h>

int main(int argc, char *argv[])
{
    int N1 = atoi(argv[1]);
    int sum1=0,prod2=1,flag1,flag2;
    int  c1_pid, c2_pid,fork1,fork2;
    fork1 = fork();
    fork2 = fork();
    if(fork1 > 0){
        for(int check=1;check <= N1;check++){
            sum1 += check;
        }
        printf("Sum is %d",sum1);
        flag1 =1;
    }
    if (fork2 > 0)
    {
        for (int check = 1; check <= N1; check++)
        {
            prod2 *= check;
        }
        printf("Product is %d", prod2);
        flag2=1;
    }
    if(flag1&&flag2==1) printf("\nDone");
    return 0;
}
```