Nmae: Shangirne Kharbanda

Registration Number: 20BAI1154

OS LAB-3

1. Create process and print parent ID and Child ID

Code:

```c
#include <stdio.h>
#include <string.h>
#include <sys/types.h>
void main()
{
Pid_t pid;
fork();
pid=getpid();
if(pid == -1)
printf("\n Error in creating process ");
else if(pid == 0)
printf("\nExecuting in child process, pid=%d and its parent pid = %d ",getpid(),getppid());
else
printf("\nExecuting in parent process,pid=%d \n",getppid());
}

```

Output:

```
alaric@alaric-virtual-machine:~/Desktop$ gedit process.c
 alaric@alaric-virtual-machine:~/Desktop$  gcc process.c
alaric@alaric-virtual-machine:~/Desktop$ ./a.out

Executing in parent process,pid=1249
```

2. Create a process and compute factorial in child and Fibonacci in parent as executable

Code:

```c
#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>
#include <sys/types.h>
#include <string.h>
#include <stdlib.h>
int main(int argc , char *argv[])
{
 int i, n;
 int t1 = 0, t2 = 1;
 int nextTerm = t1 + t2;
 printf("Fibonacci Series: %d, %d, ", t1, t2);
 for (i = 3; i <= n; ++i) {
 printf("%d, ", nextTerm);
 t1 = t2;
 t2 = nextTerm;
 nextTerm = t1 + t2;
 }
pid_t pid;

 if (argc != 2)
 {
 printf("arg missing or exceeding\n");
 exit(0);
 }

 if ( atoi ( argv[1] ) <0 )
 {
 printf("negative number entered %d", atoi(argv[1]));
 exit(0);
 }

 pid=fork();

 if ( pid<0 )
 {
 printf("failed to create child\n");
 printf("failed to create child\n");
 exit(0);
 }

 else if ( pid==0 )
 {
 int ans = 0, i, j, k = 2, n;
 n = atoi(argv[1]);
 int arr[n],sum[n];

 arr[0] = 1;
 for (i=1 ; i<n; i++)
 {
 arr[i] = arr[i-1]*k;
 k++;
 }
 for (j=0; j<n; j++)
 {
 sum[j] = 0;
 for (i=0; i<=j; i++)
 {
 printf(" %d ",arr[i]);
 sum[j]+=arr[i];
 }
 printf("\n");
 }

 for (i=0; i<n; i++)
 {
 if ((sum[i]%2) == 0)
 sum[i] = -1;
 }

 exit(0);
 }

```

```
72
73  else
74  {
75  wait(NULL);
76  }
77 }
```

Output:

```
alaric@alaric-virtual-machine:~/Desktop$ gedit process2.c
alaric@alaric-virtual-machine:~/Desktop$ gcc process2.c
alaric@alaric-virtual-machine:~/Desktop$ ./a.out 5
Fibonacci Series: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144,  1
 1  2
 1  2  6
 1  2  6  24
 1  2  6  24  120
```

3. Create a Process and let child do some task like computing sum of N numbers

Code:

```
 1 #include<stdio.h>
 2 #include<string.h>
 3 #include<sys/types.h>
 4 #include<unistd.h>
 5 int main()
 6 {
 7 id_t pid;
 8 fork();
 9 pid=getpid();
10 int n,i,sum=0;
11 printf("Enter number: ");
12 scanf("%d", &n);
13 for(i=0;i<=n;i++)
14 {
15 sum+=i;
16 }
17 printf("\nSum: %d\n",sum);
18 }
```

Output:

```
alaric@alaric-virtual-machine:~/Desktop$ gcc process3.c
alaric@alaric-virtual-machine:~/Desktop$ ./a.out
Enter number: Enter number: 15

Sum: 120
```

4. Palindrome and ODD or EVEN as parent and child with fork

Code:

```c
#include<stdio.h>
#include<string.h>
#include<sys/types.h>
#include<unistd.h>
void main()
{
int n, rev=0, rem, num;
printf("Enter number: ");
scanf("%d",&n);
num=n;
while(n!=0)
{
rem=n%10;
rev=rev*10+rem;
n/=10;
}
if(num==rev)
printf("\n%d is a palindrome\n",num);
else
printf("\n%d is not a palindrome\n",num);
fork();
if(num%2==0)
printf("\n%d is an even number\n",num);
else
printf("\n%d is a odd number\n",num);
}
```

Output:

```
alaric@alaric-virtual-machine:~/Desktop$ gcc process4.c
alaric@alaric-virtual-machine:~/Desktop$ ./a.out
Enter number: 191

191 is a palindrome

191 is a odd number
```

```
alaric@alaric-virtual-machine:~/Desktop$ gcc process4.c
alaric@alaric-virtual-machine:~/Desktop$ ./a.out
Enter number: 678

678 is not a palindrome

678 is an even number
```