NAME – ASHUTOSH ARDU

REGNO – 20BRS1262

OS LAB 2

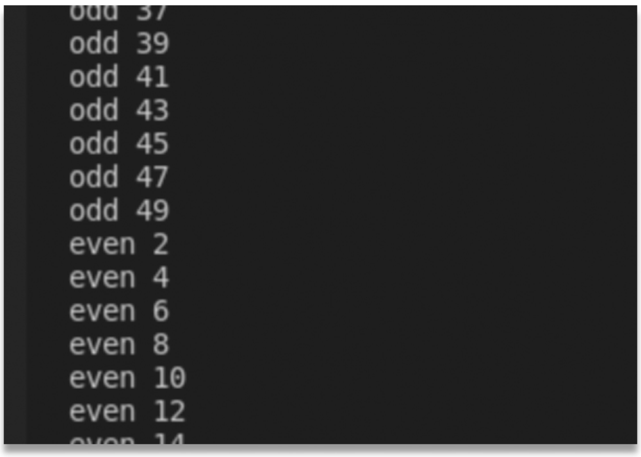
1. Write a C program to create a Child process using ***fork*** system call to print even numbers and parent prints odd number till 50.

CODE

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
int main()
{
int i;
if(fork() == 0)
{
for(i=1;i<50;i++)
if(i%2 == 0)
printf("even %d\n",i);
}
else
{
for(i=1;i<50;i++)
if(i%2 != 0)
printf("odd %d\n",i);
}
}
```

OUTPUT

A GLIMPSE OF THE ENTIRE OUTPUT



```
odd 37
odd 39
odd 41
odd 43
odd 45
odd 47
odd 49
even 2
even 4
even 6
even 8
even 10
even 12
even 14
```

2. The parent and child are two processes; can we say parent runs first always? If so, do scheduling and context switching have any impact on them?

- Between the two processes parent and the child, we can't say that the parent always runs first (though we do know that it was created first). The process which runs first is depended on the scheduling algorithm and contet switching which analyizes both the processses and sorts it for effective CPU time usage. Sometimes both parent and child processses run in synchronism.

3. In multi-processor environment, how will you ensure the parent controls child process, how will you ensure child finishes first and then parent?

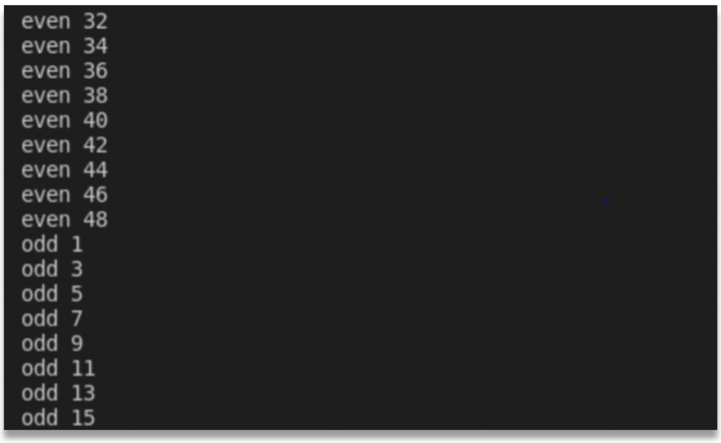
- We can ensure that child finishes before termination of the parent by simply waiting the parent process before all its child processes terminates then atlast we can terminate the parent process. The process waits allows the parent process to control the child process.

CODE

↓

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<sys/wait.h>
int main()
{
    int i;
    if(fork() == 0)
    {
    for(i=1;i<50;i++)
    if(i%2 == 0) printf("even %d\n",i);
    exit(0);
    }
    else{
        wait(0);
        for(i=1;i<50;i++)
        if(i%2 != 0) printf("odd %d\n",i);
    }
}
```

GLIMPSE OF THE OUTPUT



```
even 32
even 34
even 36
even 38
even 40
even 42
even 44
even 46
even 48
odd 1
odd 3
odd 5
odd 7
odd 9
odd 11
odd 13
odd 15
```

4. Create process and print parent ID and Child ID.

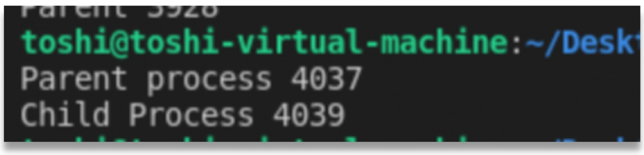
CODE

↓

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

int main(){
    pid_t c;
    if(fork()==0) exit(0);
    else{
        c=wait(0);
        printf("Parent process %d\nChild Process %d\
n",getpid(),c);
    }
    return 0;
}
```

OUTPUT



```
Parent 3928
toshi@toshi-virtual-machine:~/Desk
Parent process 4037
Child Process 4039
```

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

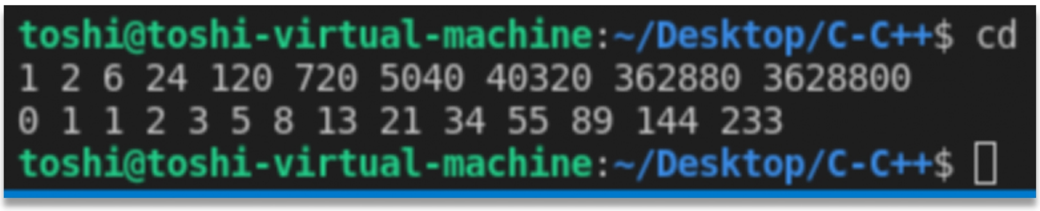
CODE

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

```c
typedef long long int ll;

ll fact(ll n){
    if(n<=1) return 1;
    else return n*fact(n-1);
}
ll fibo(ll n){
    if(n==1) return 0;
    else if(n==2) return 1;
    else if(n>=3) return fibo(n-1)+fibo(n-2);
}
int main()
{
    ll i;
    if(fork() == 0)
    {
    for(ll i=1;i<=10;++i) printf("%lld ",fact(i));
    printf("\n");
    exit(0);
    }
    else{
        wait(0);
        for(i=1;i<15;i++) printf("%lld ",fibo(i));
        printf("\n");
    }
}
```

OUTPUT



```
toshi@toshi-virtual-machine:~/Desktop/C-C++$ cd
1 2 6 24 120 720 5040 40320 362880 3628800
0 1 1 2 3 5 8 13 21 34 55 89 144 233
toshi@toshi-virtual-machine:~/Desktop/C-C++$ 
```