WEEK 2

Parent Process Computes the SUM OF EVEN and Child Process Computes the sum of ODD NUMBERS using fork

Here,we are using the concept of fork. When a child process is created, then sum of even and odd numbers in an array is calculated by child and parent process separately.

First of all, we declare an array which takes integer values. Then we enter the elements in this array. The child process is created by calling fork() system call .The returned value of fork in child is 0 whereas in parent ,it returns the PID of child process. Then we check the condition **(if (pid==0))** ,if it is true, it means child process is created. Now child process calculate sum of even numbers and parent calculate sum of odd numbers.

SYSTEM CALL USED:

fork();

System call fork() is used to create processes. It takes no arguments and returns a process ID. The purpose of fork() is to create a new process, which becomes the child process of the caller.

After a new child process is created, both processes will execute the next instruction following the fork() system call .This can be done by testing the returned value of fork().

The fork operation creates a separate address space for the child. The child process has an exact copy of all the memory segments of the parent process, though if **copy-on-write** semantics are implemented actual physical memory may not be assigned (i.e., both processes may share the same physical memory segments for a while). Both the parent and child processes possess the same code segments, but execute independently of each other.

exit()

ISOC defines exit to provide a way for a process to terminate without running exit handler or signal handlers. Whether or not standard I/O streams are flushed depends on the implementation.

PROGRAM:

```
#include<stdio.h>                                   //HEADER FILES
#include<unistd.h>
#include<stdlib.h>
#include<sys/types.h>
#include<sys/wait.h>
#define max 20                                      //SYMBOLIC CONSTANTS
int main()                                          //MAIN FUNCTION
{
pid_t pid;
int a[max],n,sum=0,i,status;
printf("\nEnter the no of terms in the array :");
scanf("%d",&n);
printf("\nEnter values in the array : ");
for(i=0;i<n;i++)                                    //LOOP
{
        scanf("%d",&a[i]);
        pid=fork();
        wait(&status);                              //WAITING FOR STATUS
        if(pid==0)
        {
                for(i=0;i<n;i++)                    //LOOP
                {
                        if(a[i]%2==0)
                        {

                        }
                        printf("Sum of even nos = %d\n",sum);

                }
                exit(0);

        }

        else
        {
                for(i=0;i<n;i++)                    //LOOP
                {
                        if(a[i]%2!=0)
                        {
                                sum=sum+a[i];       //CALCULATING SUM OF ODD
                        }
                        printf("Sum of odd nos = %d\n",sum);
                }
        }
```

```
    }
return 0;
}                                                    //END OF MAIN
```

OUTPUT:

```
Enter the no of terms in the array :6

Enter values in the array : 1 2 3 4 5 6
Sum of even nos = 12
Sum of odd nos = 9
```