

```
/* Process life cycle
```

You can specify the number of child processes to be created

Always half of the children will exit normally and the other half will exit abnormally

The children will also print their exit status though it is not needed as even if the children did not print it

the parent process will wait until all the children have executed and will also print their exit status */

```
#include<stdio.h>
```

```
#include<unistd.h>
```

```
#include<signal.h>
```

```
#include<sys/wait.h>
```

```
main()
```

```
{
```

```
    int n,i,status;
```

```
    char x[10];
```

```
    printf("\nHow many child processes do you want to create?:");
```

```
    scanf("%d",&n);
```

```
    int pid[n]; //To store process ID's of all children created
```

```
    for(i=0;i<n;i++)
```

```
    {
```

```
        pid[i]=fork();
```

```
        if(pid[i]==0) //Only the child created in each iteration can access this
```

```
        {
```

```
            sprintf(x,"%d",i); //x[i] now contains i so that each child is passed  
            a different value
```

```
            execl("/cygdrive/c/users/pushpinder/desktop/child.exe","child.exe",x,  
                NULL);
```

```
        }
```

```
    }
```

```
    sleep(3); //This is just to make the out more readable,
```

```
            //not needed to make the parent wait for all the children to finish,
```

```
            //you will get the same output even if you remove this
```

```
    for(i=0;i<n;i++)
```

```
    {
```

```
        while(waitpid(pid[i],&status,0)>0) //to make the parent keep waiting till  
        the last child is executed
```

```
        {
```

```
            // not an optimal solution as
```

```
            waitpid() may return -1 for some unforeseen situations
```

```
            // but in this simple program,it will  
            return -1 only when all children have  
            executed
```

```
        if(WIFEXITED(status)) //processes which exited normally
```

```
            printf("\nParent says:Child %d exited normally",pid[i]);
```

```
        else //aborted processes
```

```
            printf("\nParent says:Child %d was aborted",pid[i]);
```

```
        }
```

```
    }
```

```
}
```

```
/* Code for child process
```

```
The child processes will catch their own termination signals and display them  
Value passed by the parent is used to determine normal or abnormal termination */
```

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

```
void term() //Used to handle normal termination  
{  
    printf("\nChild %d says:Normal Termination\n",getpid());  
}
```

```
void abort_handler(int sig) //Used to handle SIGABRT  
{  
    printf("\nChild %d says:Abnormal Termination\n",getpid());  
}
```

```
void alarm_handler(int sig) //Used to handle SIGALRM  
{  
    printf("\nChild %d says:I am waking up\n ",getpid());  
}
```

```
main(int argc,char *argv[])  
{  
    int n=atoi(argv[1]); //n contains the value passed by the parent  
  
    printf("\nChild %d says:My parent is %d \n",getpid(),getppid());  
    printf("\nChild %d says:I am going to sleep\n",getpid());  
  
    atexit(term); //Catching normal exit and calling function term()  
    signal(SIGALRM,alarm_handler); //Catching SIGALRM  
    signal(SIGABRT,abort_handler); //Catching SIGABRT  
  
    alarm(3); //Setting the alarm  
    pause(); //Self suspension  
  
    if(n%2==0) //If even number passed by parent  
        abort();//abnormal termination  
    else //Odd number passed by parent  
        exit(0); //normal termination  
}
```

pushpinder@pushpinder-PC /cygdrive/c/users/pushpinder/Desktop

\$./parent

How many child processes do you want to create?:4

Child 5744 says:My parent is 12700

Child 5744 says:I am going to sleep

Child 10972 says:My parent is 12700

Child 10972 says:I am going to sleep

Child 1984 says:My parent is 12700

Child 1984 says:I am going to sleep

Child 14144 says:My parent is 12700

Child 14144 says:I am going to sleep

Child 5744 says:I am waking up

Child 5744 says:Abnormal Termination

Child 10972 says:I am waking up

Child 10972 says:Normal Termination

Child 1984 says:I am waking up

Child 1984 says:Abnormal Termination

Child 14144 says:I am waking up

Child 14144 says:Normal Termination

Parent says:Child 5744 was aborted

Parent says:Child 10972 exited normally

Parent says:Child 1984 was aborted

Parent says:Child 14144 exited normally

pushpinder@pushpinder-PC /cygdrive/c/users/pushpinder/Desktop

\$ |