**ASSIGNMENT NO.2**

**NAME: Sayali Kshirsagar**

**ROLLNO: TI062**

**QUESTION**: Implementthe c program in which the main program accepts an integer array. Main code uses the FORK system call to create a new process called a child process. Parent process sorts an integer array and passes the sorted array to child process through command line arguments of EXECVE system call. The child process uses EXECVE system call to load new program that uses this sorted array for the particular item in the array.

**PART 1**

#include<sys/types.h>

#include<unistd.h>

#include<stdio.h>

#include<sys/wait.h>

#include<stdlib.h>

void main(int argc,char \*argv[])

{

pid\_t pid;

int status;

int i,j=0;

int a[argc-2];

char \*args[argc];

int temp;

for(i=1;i<argc;i++,j++)//reading

args[j]=argv[i];

args[j]=0;

for(i=2,j=0;i<argc;i++,j++) //string to integer

a[j]= atoi(argv[i]);

for(i=0;i<argc-2;i++) //sorting

{

for(j=i+1;j<argc-2;j++)

{

if(a[i]>a[j])

{

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

}

printf("\n sorted numbers are::");//sorted elements

for(i=0;i<argc-2;i++)

printf("\t%d",a[i]);

for(i=0,j=1;i<argc-2;i++,j++)//integer to string

sprintf(args[j],"%d",a[i]);

fflush(stdout);

pid=fork();

if(pid==-1)

{

printf("\nERROR");

}

else if(pid==0)

{

printf("\n-------------");

execv(args[0],args);

printf("\nThis is child process with id: %d",getpid());

printf("\nParent process id: %d",getppid());

printf("\n-------------");

}

else

{

//sleep(1);

//system("ps");

wait(&status);

//system("ps");

printf("\n-------------");

printf("\nThis is parent process with id: %d",getpid());

printf("\nChild process id::%d",pid);

printf("\n-------------");

}

}

**PART 2**

#include<stdio.h>

#include<unistd.h>

#include<stdlib.h>

void main(int argc,char \*argv[])

{

int i,j=0;

int a[argc-1];

int temp;

int search;

for(i=1;i<argc;i++,j++)

a[j]=atoi(argv[i]);

printf("\n the child id id:%d",getpid());

printf("\n the parent id id:%d",getppid());

printf("\n the numbers are::");

for(i=0;i<argc-1;i++)

printf("\t%d",a[i]);

printf("\n enter the number which you want to search:");

scanf("%d",&search);

int first=0;

int last=argc-1;

int mid=(first+last)/2;

while(first<=last)

{

if(a[mid] ==search)

{

printf("\n %d is present at index %d ",search,mid+1);

break;

}

else if(a[mid] < search)

first=mid+1;

else

last=mid-1;

mid=(first+last)/2;

}

if(first>last)

printf("\n number not found");

}

**OUTPUT**

sayali@sayaliubuntu-VirtualBox:~/Desktop/OS codes$ gcc ass2part2.c -o b.out

sayali@sayaliubuntu-VirtualBox:~/Desktop/OS codes$ gcc ass2part1.c

sayali@sayaliubuntu-VirtualBox:~/Desktop/OS codes$ ./a.out ./b.out 8 6 5 7 4 9 2

sorted numbers are:: 2 4 5 6 7 8 9

the child id id:3197

the parent id id:3196

the numbers are:: 2 4 5 6 7 8 9

enter the number which you want to search:7

7 is present at index 5

-------------

This is parent process with id: 3196

Child process id::3197

-------------sayali@sayaliubuntu-VirtualBox:~/Desktop/OS codes$ ./a.out ./b.out 8 6 5 7 4 9 2

sorted numbers are:: 2 4 5 6 7 8 9

the child id id:3203

the parent id id:3202

the numbers are:: 2 4 5 6 7 8 9

enter the number which you want to search:4

4 is present at index 2

-------------

This is parent process with id: 3202

Child process id::3203

-------------sayali@sayaliubuntu-VirtualBox:~/Desktop/OS codes$ ^C

sayali@sayaliubuntu-VirtualBox:~/Desktop/OS codes$