#include <stdio.h>

#include <sys/types.h>

#include <unistd.h> //for getpid and getppid inbuilt functions

#include <stdlib.h>

#include <sys/wait.h>

void bubblesort(int \*arr,int n) //bubble sort for a child process

{

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

{

for(int j=0;j<n-1;j++)

{

if(arr[j]>arr[j+1])

{

int temp=arr[j];

arr[j]=arr[j+1];

arr[j+1]=temp;

}

}

}

}

void insertionsort(int \*arr,int n) //insertion sort for a child process

{

int i, j, temp;

for (i = 1; i < n; i++)

{

temp = arr[i];

j = i - 1;

while(j>=0 && temp <= arr[j])

{

arr[j+1] = arr[j];

j = j-1;

}

arr[j+1] = temp;

}

}

void fork1()

{

int arr[25],arr1[25],n,status;

printf("\n enter the no of integers to be sorted:");

scanf("%d",&n);

printf("\n enter the integers:");

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

{

scanf("%d",&arr[i]);

}

int pid =fork(); //fork process for creating a new child process

if(pid==0)

{

//sleep(10); //orphan process

printf("\n child process\n");

printf("\nchild process id:%d\n",getpid()); //getting child process id (getpid is an inbuilt function)

printf("\nparent process id:%d\n",getppid()); //getting parent process id (get ppid is an inbuilt function)

insertionsort(arr,n);

printf("\nelements sorted using insertion sort:");

printf("\n");

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

{

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

}

printf("\nparent process id:%d\n",getppid()); //printing parent process id

system("ps -x"); //gives a snapshot of currently executing process

}

else if(pid>0)

{

sleep(10); //zombie process

wait(NULL); //waits for the completion of child process

printf("\nparent process\n");

printf("\nparent process id=%d\n",getpid()); //get parent's process id

bubblesort(arr,n);

printf("Elements Sorted using bubblesort:");

printf("\n");

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

{

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

}

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

}

else

{

printf("Process not created successfully");

}

}

int main()

{

fork1();

return 0;

}

/\*OUTPUT

ZOMBIE PROCESS:

comp68@comp68:~/Desktop$ ps -e -o pid,ppid,stat

PID PPID STAT

7458 7326 S+

7463 7458 Z+

comp68@comp68:~/Desktop$ gcc os2.c -o os2.out

comp68@comp68:~/Desktop$ ./os2.out

enter the no of integers to be sorted:5

enter the integers:2 33 5 1 6

child process

child process id:7463

parent process id:7458

elements sorted using insertion sort:

1 2 5 6 33

parent process id:7458

ORPHAN PROCESS:

comp68@comp68:~$ cd Desktop

comp68@comp68:~/Desktop$ gcc os2.c -o os2.out

comp68@comp68:~/Desktop$ ./os2.out

enter the no of integers to be sorted:5

enter the integers:11 77 34 5 90

child process

child process id:5860

parent process id:5854

elements sorted using insertion sort:

5 11 34 77 90

parent process id:5854

parent process

parent process id=5854

Elements Sorted using bubblesort:

5 11 34 77 90

comp68@comp68:~/Desktop$ ps -e -o pid,ppid,stat

PID PPID STAT

5854 5775 S+

\*/