CSE-2008 Assignment

Academic year: 2020-2021 **Semester:** WIN

Faculty Name: Mr. Sanket Mishra sir **Date:** 01 /11/2021

Student name: M.Taran **Reg. no.:** 19BCE7346

QUESTION:

Write a program in which the parent creates 22 child processes.

The child process should print its pid to the standard output and then finish.

The parent process also needs to print its pid and terminate soon after.

a. Which process executed first, child or parent ?

b. Fetch the pid of child processes created and parent from the terminal using appropriate options with ps.

c. Is the pattern of execution in parent and child maintained ?

Program :

```c
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>
/* Creating task A */
void taskA() {
/* For making 22 processes */
int n = 22;
for(int i=0;i<n;i++)
{
if(fork() == 0)
{
```

```
printf("pid's of child processes %d\n",getpid());
exit(0);
}
printf("pid's parent processes %d\n",getpid());
}
sleep(1000);
for(int j=0;j<n;j++)
wait(NULL);
}
int main(void) {
taskA();
return EXIT_SUCCESS;
}
```

OUTPUT :

```
vitap@vitap-OptiPlex-3070:~$ cd Desktop
vitap@vitap-OptiPlex-3070:~/Desktop$ gcc -o a a.c
vitap@vitap-OptiPlex-3070:~/Desktop$ ./a
parent process = 4486/nchild processes = 4487/nparent process = 4486/nchild proc
esses = 4488/nparent process = 4486/nchild processes = 4489/nparent process = 44
86/nchild processes = 4490/nparent process = 4486/nchild processes = 4491/naren
t process = 4486/nchild processes = 4492/nparent process = 4486/nchild processes
 = 4493/nparent process = 4486/nchild processes = 4494/nparent process = 4486/nc
vitap@vitap-OptiPlex-3070:~/Desktop$
```

QUESTION:

Q2.Implement the above code using pid to decide: array = [17,16,14,7,8,9,1,2]

a. if pid == 0, implement binary search on the given set of numbers else implement linear search on a given set of numbers.

b. What is the pid of a child and parent ?

c. which executed first , linear or binary search ?

Program :

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

```c
#include <stdlib.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>
int arr[100];
void swap(int *no1, int *no2)
{
int temp = *no1;
*no1 = *no2;
*no2 = temp;
}
int getTheArray(){
int max, i;
printf("\nEnter number of inputs : ");
scanf("%d", &max);
printf("\nEnter values :");
for(i=0;i<max;i++){
scanf("%d", &arr[i]);
}
printf("\nUnsorted array is : ");
for(i=0;i<max;i++){
printf("\t%d",arr[i]);
}
printf("\n");
return max;
}
void bubblesort_parent(int n){
int i, j;
for(i=0;i<n-1;i++){
for(j=0;j<n-i-1;j++){
if (arr[j] > arr[j+1])
swap(&arr[j], &arr[j+1]);
}
}
printf("\nsorted array is : ");
for(i=0;i<n;i++){
printf("\t%d",arr[i]);
}
printf("\n");
}
void insertion_child(int n)
{
```

```
int i, key, j;
for (i = 1; i < n; i++)
{
key = arr[i];
j = i-1;
while (j >= 0 && arr[j] > key)
{
arr[j+1] = arr[j];
j = j-1;
}
arr[j+1] = key;
}
printf("\nsorted array is : ");
for(i=0;i<n;i++){
printf("\t%d",arr[i]);
}
printf("\n");
}
void binarySearch_Child(int size, int l, int h, int x)
{
if(l <= h)
{
int mid = l + (h - l) / 2;
if(arr[mid] == x)
{
printf("%d is present at index %d", x, mid);
return;
}
else if(arr[mid] > x)
{
binarySearch_Child(size, l, mid-1, x);
}
else
{
binarySearch_Child(size, mid+1, h, x);
}
}
printf("%d is not found in the array", x);
}
void LinearSearch_Parent(int n, int x)
{
for(int i=0;i<n;i++)
```

```
{
if(arr[i] == x)
{
printf("%d is present at index %d\n", x, i);
break;
}
}
}
void main() {
int n;
n=getTheArray();
bubblesort_parent(n);
if(fork()==0){
wait(NULL);
printf("\nChild : ");
int x;
printf("\nEnter the value to be searched : ");
scanf("%d", &x);
binarySearch_Child(n, 0, n-1, x);
}else{
wait(NULL);
printf("\nParent Process : ");
int x;
printf("\nEnter the value to be searched : ");
scanf("%d", &x);
LinearSearch_Parent(n, x);
}
}
```

OUTPUT :