CED19I056

Venkata Sai Nandita M

Ques1:

```
nandita@DESKTOP-2LH63U6:~/OSLab1/OSLab4$ cd "/home/nandita/OSLab1/OSLab4/"
Pid=1027 PPid=1026
Pid=1028 PPid=1026
Pid=1031 PPid=1028
Pid=1029 PPid=1027
Pid=1030 PPid=1026
Pid=1026 PPid=178
Pid=1033 PPid=1029
Pid=1032 PPid=1027
```

Ques2:

Content of directory

```
nandita@DESKTOP-2LH63U6:~/OSLab1/OSLab4$ cd "/home/nandita/OSLab1/OSLab4/" && gcc Ques2.c -o Ques2 && "/home/nandita/OSLab1/OSLab4/"Ques2
Ques1    Ques2    Ques3.1    Ques3.2    Ques3.3    Ques3.4    Ques4    Ques5    tempCodeRunnerFile.c
Ques1.c  Ques2.c  Ques3.1.c  Ques3.2.c  Ques3.3.c  Ques3.4.c  Ques4.c  Ques5.c
nandita@DESKTOP-2LH63U6:~/OSLab1/OSLab4$
```

Process tree

```
nandita@DESKTOP-2LH63U6:~/OSLab1/OSLab4$ cd "/home/nandita/OSLab1/OSLab4/" && gcc Ques2.c -o Ques2 && "/home/nandita/O
init─┬─init───init───sh───sh───sh───node─┬─node─┬─bash
     │                                    │      ├─bash───pstree
     │                                    │      └─12*[{node}]
     │                                    ├─node───12*[{node}]
     │                                    ├─node───10*[{node}]
     │                                    └─10*[{node}]
     └─{init}
```

Ques3.1:

- execl()

```
nandita@DESKTOP-2LH63U6:~/OSLab1/OSLab4$ cd "/home/nandita/OSLab1/OSLab4/" && gcc Ques3.1.c -o Ques3.1
total 4.0K
drwxr-xr-x 14 nandita nandita 4.0K Aug 29 13:35 nandita
```

- execlp()

```
nandita@DESKTOP-2LH63U6:~/OSLab1/OSLab4$ cd "/home/nandita/OSLab1/OSLab4/" && gcc Ques3.5.c -o Ques3.5
total 4.0K
drwxr-xr-x 14 nandita nandita 4.0K Aug 29 13:35 nandita
```

- execv()

```
nandita@DESKTOP-2LH63U6:~/OSLab1/OSLab4$ cd "/home/nandita/OSLab1
total 4.0K
drwxr-xr-x 14 nandita nandita 4.0K Aug 29 13:35 nandita
```

- execvp()

```
nandita@DESKTOP-2LH63U6:~/OSLab1/OSLab4$ cd "/home/nandita/OSLab1/OSLab4/" && gcc Ques3.6.c -o Ques3.6 
total 4.0K
drwxr-xr-x 14 nandita nandita 4.0K Aug 29 13:35 nandita
```

- Wait:

```
nandita@DESKTOP-2LH63U6:~/OSLab1/OSLab4$ cd "/home/nandita/OSLab1/OSLab4/" &&
Exit status: 1
```

Ques4:

```
nandita@DESKTOP-2LH63U6:~/OSLab1/OSLab4$ cd "/home/nandi
Enter the size: 15
Parent Process
Even sum is 56
Child Process
Odd sum is 64
```

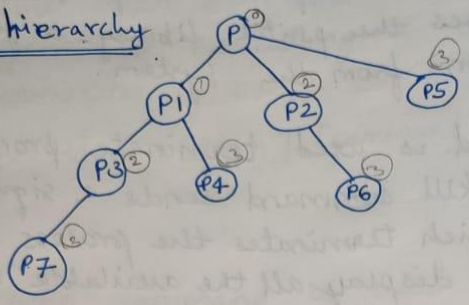
Ques5:

```
nandita@DESKTOP-2LH63U6:~/OSLab1/OSLab4$ cd "/home/nandita/OSLab1/OSLab4/" &

Enter the size of array :10

Enter the elements:4
6
7
9
3
2
5
6
7
8

Child processThe numbers in Descending order are:
9       8       7       7       6       6       5       4       3       2

Parent processThe numbers in Ascending order are:
2       3       4       5       6       6       7       7       8       9
```

M. Nandita
CED19I056

1) Program to create multiple processes using fork.

Tree hierarchy



Total number of processes = $2^n = 2^3 = 8$.

4) There we are supposed to find the odd sum and even sum.
So, what I have done is forked the process and ~~if~~ in the parent process I have counted the sum of even numbers. and in the child process I have counted the sum of odd numbers.

5) Here we are to ~~dis~~ sort the numbers in asending and decending order. So if the pid is >0 then ~~it~~ it is a Parent process so, here we sort the numbers in acending order and if pid =0 then it is a child process so we sort the numbers in decending order. Here so that child is completed before parent ~~I have added sleep(5) else the p>0~~. I have used waitpid() so that the parent process waits for the child process to complete.

Ques 3) 1. e
f
c
2.
3
A
6

Ques 3) 1. execl()
first arg, second arg are path of executable
arg to path followed by NULL.

2. execlp()
full path is not required, filename is enough.

3. execv()
Same as execl() by this one is a
null terminated array.

4. execvp()
full path is not required, filename is enough.
it is null terminated array.

5. wait(&stat)
on success, returns the process Id of the terminated child, on error -1 is returned

6. waitpid():
This system call suspends the execution of the current process untill a child specified by pid argument has changed state.

Tag

<-1	wait for any child process grp id is equal to abs value of pid
-1	meaning wait for any child process.
0	wait for the child process whose process id is same as groupid.

wifexited(status) returns true if the child terminated normally.

wifesignaled(status) returns true if the child proces was terminated by a signal.

wtermsig(status) returns the number of the signal that caused the child proces to terminate.

2. Here I have used ls to display the content of the directory and pstree to show the process tree.