SP Assignment

Q1(Quiz # 1, Section A): Write a program that searches for a file passed to it as a command line argument in all the provided paths. Take paths as CLA.

Sample Run:

./find . .. ~/Desktop

Q2(Quiz # 1, Section B): Write a program to implement ls command. Take the name of the directory to be listed from command line. Also print the path of CWD.

Sample Run:

./t1.o SP

Q3(Quiz # 1, Section C): Write a program that finds a file in a directory. Program shall receive the name of the file & directory from command line.

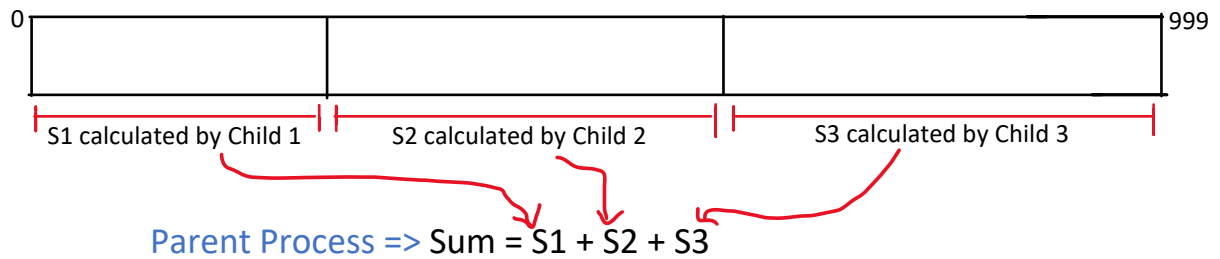**Sample Run:**

./find.o SP task1.c

Q4(Quiz # 2, Section A): Write a program that implements FTP Server. Client requests for the contents of a specific directory. Server responds with the list of files/directories.

Q5(Quiz # 2, Section B): Write a program that implements a simple FTP Server. Client requests for a file and server responds with the contents of the file. Client shall receive the contents and display on STD_OUT.

Q6(Quiz # 2, Section C): Write a program for continuous communication (2-Way) between parent & child process using pipes.

Q7(Quiz # 3, Section A,B,C): Write a program for parallel array addition. The program must create 3 child processes and each child should calculate the sum of the one-third (1/3) of array elements. Parent process shall receive the sum calculated by each child, add them to get final sum and then display it. Make sure there are no orphan child processes.

You can use pipes, fifos or return value of child processes for Inter Process Communication.

```
0                                                                    999
┌──────────────────┬──────────────────┬──────────────────────────────┐
│                  │                  │                              │
└──────────────────┴──────────────────┴──────────────────────────────┘
  S1 calculated by Child 1   S2 calculated by Child 2   S3 calculated by Child 3
```

Parent Process => Sum = S1 + S2 + S3

Q8(Quiz # 4, Section A, C): Write a program that creates a child process. Child process shall send "N" SIGUSR1 or SIGUSR2 to parent process. Parent process shall count the number of SIGUSR2 received.

Q9(Quiz # 4, Section B): Write a program that creates a child process & waits for the child process to terminate using **pause/sigsuspend/sigwait.**

Q10(Quiz # 5, Section B): Write a program that creates 2 threads.

Thread 1: Find sum of array elements.

Thread 2: Searches for a key in array.

Q11(Quiz # 5, Section C): Write a multithreaded program for parallel file copying. Open both source files in master thread before creating threads.

Thread 1	Thread 2
S1 → D1	S2 → D2