**Operating Systems Lab 2021**

**Assignment Set 2:**

**Submission Deadline 11th Sept 2021**

**Objectives: to learn the following:**

- **singal() system call**
- **kill() system call**
- **exit() system call**
- **wait() system call**
- **waitpid() system call**
- **sleep() system call**
- **application using SICHLD, SIGINT, SIGTSTP, etc. signals**

---

A1) Write a code to speedup search in a large unsorted integer array by creating (n) child processes each to search the array just to know if the element is found in the array. Each child should search a non-overlapping portion of the array. The parent takes input to the array elements randomly and asks the user to input the element to be searched. The search process should stop as soon as any one of the child finds a success. The child process exits with status code(1) if successful. Use wait() system call for parent child synchronization.

==========

A2) A program where a parent forks (n) child processes to speedup finding all prime numbers in some given range [x,y], where both x & y are provided as input and both are > 1000. Each child writes the prime numbers found in a common file which the parent should create before creating any of the child process. The parent suspends it till all the child processes completes their execution. Use wait() system call for parent child synchronization.

==========

A3) Solve Problem (A1) using waitpid() system call for parent child synchronization.

==========

A4) Solve Problem (A2) using waitpid() system call for parent child synchronization.

==========

A5) Write a code where the parent creates a child using fork(). The following need to be done:

==========

The child sleeps for 5 secs, generates a random number (x) 0 or 1, displays “Child: Hello” if 0 else displays “Child: World” and terminates calling exit(x).

The parent synchronizes with the child calling the wait() system call and displays “Parent: World” if child displayed “Child: Hello” else “Parent: World”. (The parent decodes the exit code to know what the child displayed)

A6) Write a code where the parent creates a child using fork(). The following need to be done:

The child sleeps for 5 secs, generates a random number (x) 0 or 1; displays “Child: Hello” if x is 0 and issues SIGUSR1 signal to parent ELSE if x is 1 displays “Child: World” and issues SIGUSR2 signal to parent. The child repeats the task in an infinite loop.

The parent on receipt of signal SIGUSR1 displays “Parent: World” and displays “Parent: Hello” if SIGUSR2 is received from child. Make provisions that this game continues till parent displays “Parent: Hello” 5 times and it sends a SIGTSTP to the child and itself exits.