(1) Implementation of Unix commands

Your implementation must support the following common usages:

(a) ls

- ls, ls -l, ls -a, ls -R, ls <dirname>

(b) rm

-rm <filename>, rm -i <filename>, rm <file1> <file2> ... -

(c) cat

-cat <file>, cat <file1> <file2> ..., cat

(d) kill

-kill pid, kill -<signo> <pid>, kill <pid1> <pid2> ..., kill -<signo> <pid1> <pid2> ...

(e) ps

-ps, ps -a, ps -ae, ps -u <username>

(f) wc

-wc <file1> <file2> ..., wc -c <file1> <file2> ..., wc -l <file1> <file2> ..., wc -w <file1> <file2> ...

(2) Command interpreter

- Display a command prompt ($)

- Read a line of command

- Divide that into separate words (arguments)

- Determine if the command name is absolute path name, if not searching the path find absolute path name for the given command. (Optional)

- Check for any I/O redirections and background execution options, adjust the arguments list prepared

- Fork a child process

- In the child process if necessary open redirected I/O targets and duplicate the descriptors at appropriate descriptor numbers.

- Use appropriate exec system call and execute the new command.

- In the parent process if not opted for background execution then wait for the termination of child process. otherwise display the next command prompt.

- Repeat the above steps until exit command is typed.

(3) Pthread and System V IPC based solutions for four classical IPC problems.

- Practice the given pthread source programs. You can also try the examples given in the installation directory.

- After understanding the pthread library, implement solutions to four IPC problems.

- Read and understand System V IPC functions.

- After understanding the pthread library, implement solutions to four IPC problems.

- Do not use semaphores using semctl call and observing the value of semaphore.

- Remember the definition of Semaphore - "Semaphore is a tool for synchronization, that provides two atomic operations wait and signal apart from initialization". So once intialized it should be used with only wait and signal operations. Even though there are mechanisms exist for accessing the semaphore value we must use only wait and signal operations.

(6) Implement deadlock avoidance and detection algorithms and page replacement algorithms using pthreads and System V IPC functions.

Written assignment based on pthread example programs.

Execute the given reduction, sieve, quicksort, matrix mulitplication programs by varying the data size and number of threads.

Program Name : xxx

Data size	No of Threads	Execution time 1	Execution time 2	Execution time 3	Execution time 4	Average Execution time
Size 1	1					
	2					
	4					
	8					
Size 2	1					
	2					
	4					
	8					
Size 3	1					
	2					
	4					
	8					
Size 4	1					
	2					
	4					
	8					