UDIT JAIN 2016CS10327
Report: A1

System Calls Implemented

1. Shutdown -> Shuts down xv6 inside qemu emulator by sending a signal
2. Print_count -> Prints count of calls since last toggled
3. Toggle -> Toggles System trace on or off.
4. Add -> Adds given integers
5. Ps -> Lists current processes
6. Send -> Sends a message from sender to target process. Implemented a buffer area in sender to keep messages in kernel space.
7. Recv -> A blocking system call to go to sleep if not yet any message received. And put into a waiting queue. If message already present then retrieve message.
8. Sigset -> Set the signal handler to given function
9. Sigsend -> Send a signal to another process given pid.
10. Sigret -> Returns from user_signal handler back into kernel space and unloads trapframe and moves %eip back.
11. SendMulti -> Sends a message to multiple receivers and the message is received by an interrupt.

Distributed Algorithm:

Store array from the file into global memory and partition that shared memory according to child numbers. Receive and add the total from each child process and terminate.
Send mean to each child process again and ask each child to calculate (x[i] -u) ^2. Receive in parent and add again and divide by total to get real variance.

IPC:

Add 3 system calls to set signal handler of the process, invoke it and return from it respectively.
Auxiliary functions in proc.c and sysproc.c
Awake the signal handler when interrupt processing pending.
Else let the receiving process sleep.