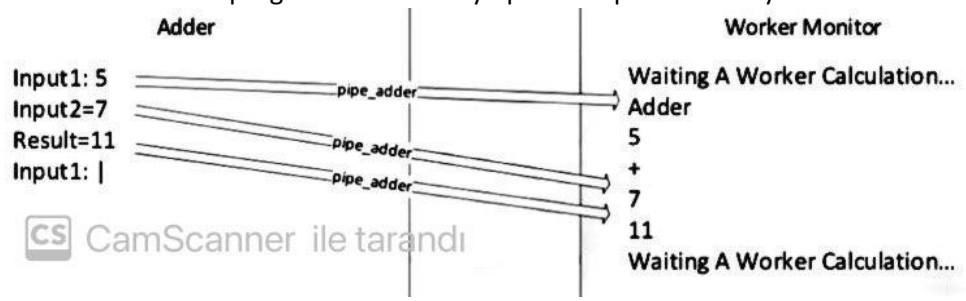
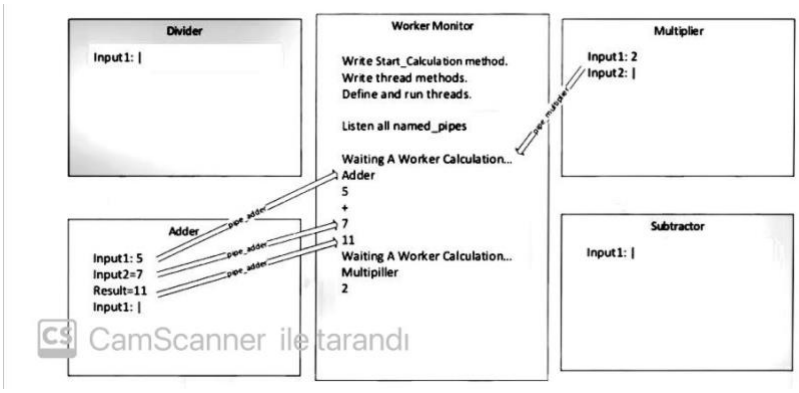
1. Prerequisites
• Must be able to use Linux based operating system.
• Must be able to use terminal.
• Must have basic knowledge of C language.
• Must be able to compile programs from terminal.
• Must be able to create processes.
• Must be able to make program calls.
• Must be able to communicate between processes.
• Must be able to use threads.
• Must be able to use signals.
2. Project Name
Worker Monitoring
3. Subject
There are five programs in the project. These are adder, subtractor, multiplier, divider and worker monitor(WM).
• adder.c -> adder
• subtractor.c -> subtractor
• multiplier.c -> multiplier
• divider.c -> divider
• worker_monitor.c -> wm
Each program must run in a separate terminal. That is, each program must be called as ./program in a separate terminal. Worker programs constantly request two numbers from the keyboard and perform the corresponding operation on them.
The worker monitor program shows every operation performed by the workers on its own screen.



WM must listen to each worker through separate pipes AT THE SAME TIME. You can do this by assigning different methods to each thread(4 thread). Listening continues until WM is closed.
If a worker has started calculating, the operations of other workers are held until that worker finishes its job.You can achieve this with lock mechanisms. The Start_Calculation function is called when any worker sends a number. This function takes the first number that comes up, the transaction type and the relevant named_pipe, or you can define like this. There should be a lock mechanism in this method and only one thread should be able to enter the critical region.The second value and result of the relevant worker is read and printed on the screen, and the worker leaves the critical region and starts listening first number again. Meanwhile, if another worker thread is waiting at the lock mechanism, it takes lock and and does same presedures.



One worker should not interfere while the other is showing its operations on the monitor.

4. Project Delivery

• The project should be written in C language, taking into account the topics covered in the operating system course. You have the knowledge and research-learning ability to solve the entire project.

• Write down the report format given for project delivery PROPERLY and CAREFULLY.

• Write the project code part, following basic programming rules and including comment lines .

• Create a makefile file with the program codes (c files) included in the project. Programs should be compiled automatically when the Makefile file is run. Homework checker will just run the makefile and then run the myshell program from the terminal. Projects without a Makefile are not evaluated.

o Documents to be included in the project file to be sent: ▪ makefile

▪ adder.c -> adder

▪ subtractor.c -> subtractor

▪ multiplier.c -> multiplier ▪ divider.c -> divider

▪ worker_monitor.c -> wm