

# Project

Week 08 - Lab 8

<https://goo.gl/4WKWFw>

# To get participation grade

- Complete coding project (chose any one of 3 options)

OR

- complete theoretical exercises instead (all 4 of them)

# Project 1

Implement the following in C

- `./shell_sv <server-port>`

The server creates a listening socket that listens on the specified port and accepts client requests containing shell commands. The server concurrently handles clients, executing each client's command, and passing the results back across the client's socket.

- `./shell_cl <server-host> <server-port> 'shell command'`

The client connects to the shell server, sends it a single shell command, reads the results sent back across the socket by the server, and displays the results on stdout.

# Project 2

- Write a program that simulates a paging system using the aging algorithm. The number of page frames is a parameter. The sequence of page references should be read from a file. For a given input file, plot the number of page faults per 1000 memory references as a function of the number of page frames available.

# Project 2 cont.

- Input:
  - Name of the file which contains a sequence of strings representing page reference.
  - Frame size.
- Output :
  - String which represents the state for each frame (hit or miss)
  - A number representing hit/miss ratio

# Project 3

- Write C implementation of Dining Philosophers problem and it's solution. First, your program should demonstrate the possibility of a deadlock situation. Second, your program should demonstrate how such situation can be resolved by using thread synchronisation means (mutex, semaphore etc.)

## Project 3 (cont.)

There are 5 philosophers sitting at a round table. Between each adjacent pair of philosophers is a chopstick. In other words, there are five chopsticks. Each philosopher does two things: think and eat. The philosopher thinks for a while, and then stops thinking and becomes hungry. When the philosopher becomes hungry, he/she cannot eat until he/she owns the chopsticks to his/her left and right. When the philosopher is done eating he/she puts down the chopsticks and begins thinking again.

# Grading criteria (projects)

- 2 points - your project executable runs and demonstrates what is asked for in the description. Source code is clear, well commented and not plagiarised
- 1 point - a strong attempt to implement the project has been made. Source code contains overall logic to implement what is asked in the description
- 0 points - otherwise



# Theoretical exercises

1. To a programmer, a system call looks like any other call to a library procedure. Is it important that a programmer know which library procedures result in system calls? Under what circumstances and why?
2. Isolation of one process from another is provided by virtual memory. Running 2 operating systems concurrently can lead to memory management difficulties. Explain what kind of difficulties can occur. How might these difficulties be addressed?
3. Consider a computer system that has cache memory, main memory (RAM) and disk, and an operating system that uses virtual memory. It takes 2 nsec to access a word from the cache, 15 nsec to access a word from the RAM, and 5 ms to access a word from the disk. If the cache hit rate is 80% and main memory hit rate (after a cache miss) is 99%, what is the average time to access a word?
4. The amount of disk space that must be available for page storage is related to the maximum number of processes,  $n$ , the number of bytes in the virtual address space,  $v$ , and the number of bytes of RAM,  $r$ . Give an expression for the worst-case disk-space requirements. How realistic is this amount?