Introduction

Week 01 – Tutorial

Chapter 1.1-1.4 & 1.8, C

Tutorial’s goal

Help you learn about OS

by solving problems

Outcomes

- You will:
  - learn how to approach problems
  - solve couple of problems
  - construct a problem for your classmate
- Today we work in pairs

Part 1: Solve a problem

Exercise #1

- First, you have to find a peer
- **Hint:** a neighbour…

Problem 1.1

- What are the two main functions of an operating system?

- **Hint.** You should know the answer !

Problem 1.1 - Answer

- What are the two main functions of an operating system?
- Two main functions are:
  - Resource management
  - Provide fine abstractions from the bare metal

Problem 1.3

- What is the difference between time-sharing and multiprogramming systems?
- **Hint.** You should:
  - recall the concepts (think a bit)
  - give the answer (explain in a couple of sentences)

Problem 1.3 - Answer

- What is the difference between time-sharing and multiprogramming systems?
- Multiprogramming is the allocation of multiple programs on one computer simultaneously. Time-sharing is the sharing of computing resources among several users at the same time. All timesharing systems are multiprogramming systems but not all multiprogramming systems are timesharing systems.

Let’s try it!

- What is kernel mode?
- What is the difference between kernel and user mode?
- Explain how having two distinct modes aids in designing an operating system.

Problem 1.10 - Answer

- In Kernel mode, the executing code has complete access to the underlying hardware. It can execute any CPU instruction and reference any memory address. Kernel mode is generally reserved for the lowest-level, most trusted functions of the operating system. Crashes in kernel mode are catastrophic; they will halt the entire PC.
- In User mode, the executing code cannot directly access hardware or reference memory. Code running in user mode must delegate to system APIs to access hardware or memory.
- Due to the protection afforded by this sort of isolation, crashes in user mode are always recoverable.

Problem 1.8

- One reason GUIs were initially slow to be adopted was the cost of the hardware needed to support them.
  - How much video RAM is needed to support a 25-line × 80-row character monochrome text screen?
  - How much for a 1200 × 900-pixel 24-bit color bitmap?
  - What was the cost of this RAM at 1980 prices ($5/KB)?
  - How much is it now?

Don't panic

Hint:

- This is a “calculation question”…
  - … you need some of basic intuition (2+2)
  - … or you need a formula from the book.
- This is a “pen&pencil” question:
  - write something
  - draw something
- That’s that simple

Problem 1.8 - Answer

- A monochrome text screen requires a buffer size of: $25 \times 80 = 2000$ (byte).

  Price in 1980 is 10\$.
- The bitmap requires:

  $1200 \times 900 \times 24$ (bit) = 3,240,000 bytes.

  Price in 1980 is:

  3,240(B) / 1024×5 (\$/KB) = \$15,820.

Let's try it!

- Consider a system that has two CPUs, each CPU having two threads (hyper-threading). Suppose three programs, P0, P1, and P2, are started with run times of 5, 10 and 20 msec, respectively. How long will it take to complete the execution of these programs? Assume that all three programs are 100% CPU bound, do not block during execution, and do not change CPUs once assigned.

Problem 1.13 - Answer

- It depends how the operating system schedules them:

Problem 1.13 - Answer

- It depends how the operating system schedules them:
  - If P0 and P1 are scheduled on the same CPU and P2 is scheduled on the other CPU, it will take 20 mses

Problem 1.13 - Answer

- It depends how the operating system schedules them:
  - If P0 and P1 are scheduled on the same CPU and P2 is scheduled on the other CPU, it will take 20 mses
  - If P0 and P2 are scheduled on the same CPU and P1 is scheduled on the other CPU, it will take 25 msec

Problem 1.13 - Answer

- It depends how the operating system schedules them:
  - If P0 and P1 are scheduled on the same CPU and P2 is scheduled on the other CPU, it will take 20 mses
  - If P0 and P2 are scheduled on the same CPU and P1 is scheduled on the other CPU, it will take 25 msec
  - If P1 and P2 are scheduled on the same CPU and P0 is scheduled on the other CPU, it will take 30 msec

Problem 1.13 - Answer

- It depends how the operating system schedules them:
  – If P0 and P1 are scheduled on the same CPU and P2 is scheduled on the other CPU, it will take 20 mses
  – If P0 and P2 are scheduled on the same CPU and P1 is scheduled on the other CPU, it will take 25 msec
  – If P1 and P2 are scheduled on the same CPU and P0 is scheduled on the other CPU, it will take 30 msec
  – *If all three are on the same CPU, it will take 35 msec

Problem 1.15

- A computer system has cache memory, main memory (RAM) and disk, and an operating system that uses virtual memory.
- It takes:
  - 1 nsec to access a word from the cache,
  - 10 nsec to access a word from the RAM, and
  - 10 ms to access a word from the disk (10,000,000 nsec).
- If the cache hit rate is 95% and main memory hit rate (after a cache miss) is 99%, what is the average time to access a word?

Problem 1.15 - Answer

Average access time =

0.95×1 nsec (word is in cache) +

$0.05 \times 0.99 \times 10$ nsec (in RAM, not in cache) +

$0.05 \times 0.01 \times 10{,}000{,}000$ nsec (on disk only) =

5002.395 nsec = 5.002395 μsec

Well done!

Take a break

Part 2: Construct a problem

Guideline: work in pairs

- Open the book, Ch. 1
- Select a topic (interesting for you)
- Read text, understand it and construct:
  - 3 “what is..” problems;
  - 2 “explain / compare” problems;
  - 1 “calculation” problem.
- With answers, please…, but do not share answers yet

Game

- Ask for a problem from the ‘left’ group:
  - For a correct solution you get 1 point per solution
- Propose your problems to the ‘right’ group:
  - If a problem wasn’t solved you get 2 points per problem
- Count your points

Summary + Questions

End

Week 01 - Tutorial