Week 12 (Revision)

The main aims of this week's lecture are to:

- refresh your memory on what *Major Topics* were covered during the semester
  - a brief review of these topics
- discuss the format of the *Final Exam*:
  - what topic areas are covered (& which are examinable)
  - the type of questions to expect, or not expect

What have we learnt?

- Introduction to a ***typical*** computer system
- Computing history
- Operating Systems (O/S's)
  - File Management
  - Process Scheduling
  - Memory Management
- Unix Operating System Basics
- Unix Shell Scripting
- System Performance Enhancements & Measurements

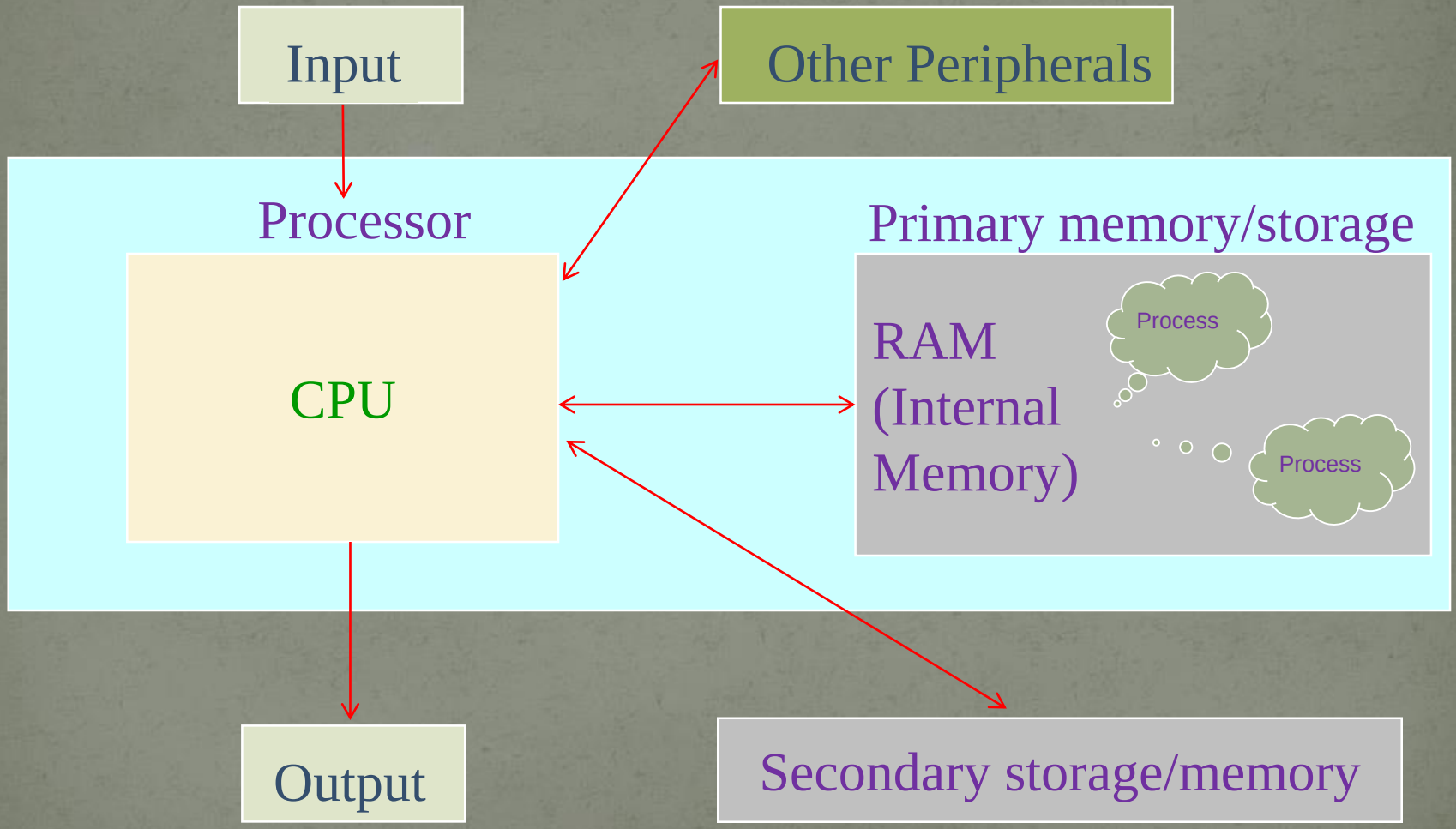
- Practical Tasks : Installing & managing/using a small ***Ubuntu*** Linux system. Installing and configuring Linux applications. Simple Unix Shell Scripting.

Topic Overview

A ***Typical Computer System*** - what are the various components in a typical computing environment?

- Hardware, software, users, processes
- The working of the idealised "Simple Digital Computer"
- CPU, Memory, Bus, Cache, I/O Devices, etc
- Secondary storages
- Fetch/execute cycle
- Machine Instructions, Assembler, High-Level Languages
- Interrupts
  - Software/Timer/I-O interrupts

Outline of a Simplified Computer



Topic Overview Cont'd

Computing History

- Computers in the early days
- Computing pioneers
- Operating Systems, programming languages

Topic Overview Cont'd

O/S Introduction

- O/S functions: file/memory/process management
- O/S types: single/multi-users, single/multi-tasking, …
- Unix, Linux, MS Windows, …

Topic Overview Cont'd

File Management

- Concept of a File
- Logical versus Physical File Systems
  - Hierarchical versus Flat file systems
- Basic File Management Features
  - mapping logical to physical file systems
  - file manipulation (creation, copy, move, etc…)
  - user management
- File access & Security

Topic Overview Cont'd

Memory Management

- Primary versus Secondary memory
- Logical versus Physical Address mapping
  - mapping logical Pages to physical Frames
- Memory Partitioning – Fixed versus Dynamic
- Memory Fragmentation – Internal versus External
- Virtual Memory Techniques
  - Paging versus Segmentation
  - different algorithms of implementation
- Program Locality and Working Set

Topic Overview Cont'd

Process Management

- Processes & Threads
- Process States (Ready/Running/Blocked) & transitions
- Process Table & Process Control Blocks
  - store info about the individual process
- Process Scheduling
  - Scheduling algorithms
- IPCs & Deadlocks
  - Deadlock conditions
  - Semaphores (*wait* & *signal*), Critical Sections
  - IPC: inter-process communication mechanisms

Topic Overview Cont'd

Unix Operating System

- Unix File types
  - ordinary files, directories, special files
  - symbolic versus hard links
  - standard input/output/error
- Unix File System (hierarchical)
  - root directory, user home directories
  - absolute versus relative paths
  - logical versus physical views
- File & Directory permissions
- I/O/Error redirections, Pipes
- Different ***Shell*** variants, eg . bash, tcsh, etc
- Unix commands

Topic Overview Cont'd

Unix Shell Scripting

- Unix commands
  - pattern matching, command sequences, command chaining/piping
- Shell Scripts
  - variables : built-in versus user-defined
  - escape character, single/double/back-quotes
  - sequence/selection/repetition constructs
  - common errors : incorrect use of spaces, wrong type of quotes, wrong syntax, using variables incorrectly, etc

Topic Overview Cont'd

Improving Technologies/Performance

- Faster processing within CPU
  - CISC/RISC architecture, faster clock speed, instruction pipelining, scaler/superscalar processing, etc
- Multi-threading (single CPU)
- Multiple CPU's (single computer)
- Distributed systems (multiple co-operating computers)
- Wider, faster data and instruction buses
- Faster external disk access
- More (and faster) memory, including advanced cache technology. Better memory managements, such as Virtual Memory
- Faster, better quality display
- Performance benchmarks

Final Exam

Here are some details for the Final Exam:

Date : Check your Monash Portal
Time : Check your Monash Portal
Duration : ***2 hours*** (+10 mins reading)
Venue : Check your Monash Portal

- paper-based; closed-book; no computer access; no calculators.
- worth 60% of subject assessment (the other 40% is from the lab session tasks).
- covers all materials from weeks 1-12 (inclusive).

Study Tips for Exam

- Here are some suggestions on how to ***prepare*** for the Final Exam:
  - Practical Questions:
    - re-do all your lab exercises, without looking at your previous answers:
    - form study groups with other students, and compare/discuss each others' solutions
  - Theory Questions:
    - read through all the lecture notes, making sure you understand all the concepts presented in them.
  - check out the general format (as shown in this lecture) of the Exam to see what type of questions to expect.

Final Exam General Format

*** Please note the following points:*

- *these are just rough guides, indicating the major topic(s) to be examined in each question – so **DO NOT** memorize the examples shown!*
- *the mark allocations are only approximates (but are usually fairly accurate).*

Final Exam General Format

- ***all*** answers are to be written on the question booklet itself. ***No*** separate answer booklet will be provided.

- blank spaces are provided on the question booklet for you to write your answers:
  - write only within the spaces provided
    - overly-long answers will not be marked
  - for the multiple-choice questions, circle your answers on the actual questions

Final Exam General Format

Section A: Multiple Choice ~20%

10 Questions:

- short questions to test your understanding of *Computer Technology* concepts & terminologies
- cover ***all*** topics (theory + practical), including basic Unix concepts/commands
- attempt all questions:
  - only mark one answer per question
  - no marks deducted for incorrect answers

Final Exam Format (contd.)

Section B: Short-Answers ~80%

5 Questions (multi-parts) :

- short-answer questions, covering:
  - theory
    - various theoretical concepts
    - do not write overly-long answers!
    - write only within the spaces provided
  - practical
    - Unix shell commands
    - Unix vi commands
    - Unix shell scripting

Final Exam Format (contd.)

Q.11 ~19%

- Unix Terminologies/Concepts
  - eg. Unix shells, command line arguments, directory structure, …
- small topics on Unix shell commands
  - eg. basic Unix shell command knowledge

Final Exam Format (contd.)

Q.12 ~17%

- General O/S Concepts
  - eg. Process/Memory/File Management, Deadlocks, IPCs, Semaphores, Critical Sections, Scheduling, Process States, etc

Q.13 ~10%

- Unix vi commands
  - commands to carry out common tasks

Final Exam Format (contd.)

Q.14 ~18%

- Small misc topics
  - eg. Unix shell concepts, file systems/paths, IP addresses, subnet masks, etc

Final Exam Format (contd.)

Q.15 ~16% (practical question)

- Unix Shell basics
  - common Unix commands to perform basic tasks
- Shell Scripting
  - basic Unix scripts
  - need to know names of common shell commands, formats of the ***sequence/selection/repetition*** constructs, general syntax, etc
  - ** ***Sufficient to base revision on lab tasks*** **

Answering exam questions

- the number of marks allocated to a question is generally a *rough* guide (or hint) to how complex the answer is… eg. a 5-mark question would probably not require a 15-page, one-hour, answer!

- write clearly and in a tidy manner. Do not use *RED* pen.

- read the question carefully – an excellent answer to the wrong question will not score any marks!

Answering exam questions…

- follow the instructions – eg. if the question asks you to describe 3 things, do not describe more than 3!

- if you cannot completely answer a question, do not just skip it entirely – answer the part(s) that you do understand, to earn at least some marks.

- do not change the question to suit your answer!

Some sample Questions…

Multiple-choice Question examples:

Q.a Which of the following statements is ***not true*** about process scheduling?

(a) pre-emptive scheduling means the O/S can force a process to relinquish the CPU
(b) a “round-robin” algorithm means processes are executed in the reverse order that they arrive
(c) a high-level scheduler decides which process will be admitted into the system ready queue
(d) both (b) and (c)
(e) both (a) and (c)

Q.b Which of the following is a characteristic of a *RISC*-based CPU architecture?

(a) small number of high-speed registers
(b) small instruction word size
(c) small number of specialised instructions
(d) none of the above

Some sample Questions…

Theory Question example:

Q.x [8+8=16 marks]

(a) Briefly explain what a ***semaphore*** is, in the context of managing non-sharable resources. Your answer must include a brief explanation of the 2 main operations allowable on semaphores.

…. Space provided for answer …. ← write only within this space!

(b) Pick any ***TWO*** (***2***) of the following Unix concepts, and explain them briefly. Provide a simple example to illustrate each of them.

i. file system hierarchy
ii. file permissions
iii. internal memory fragmentation

…. Space provided for answer …. ← write only within this space!

Some sample Questions…

Practical Question example :

Q.y [9+7=16 marks]

a) Write a single ***Unix*** command to perform each of the following tasks :

- list all the files in the **/usr/bin** directory (recursively) whose names start with '**h**', and contain at least 5 letters. Assume that you are in your home directory.

…. Space provided for answer …. ← write only within this space!

- etc...

Assume that the operating system in question is ***Ubuntu Linux***, and the shell used is ***bash***.
Unless stated otherwise, only one single command is needed for each task.
The tasks are not related to each other.

b) Write a script to perform the following tasks……
(Similar to the scripting questions from the Week 10-12 labs. Read the questions carefully!)

…. Space provided for answer …. ← write only within this space!

Best of luck for all your exams!