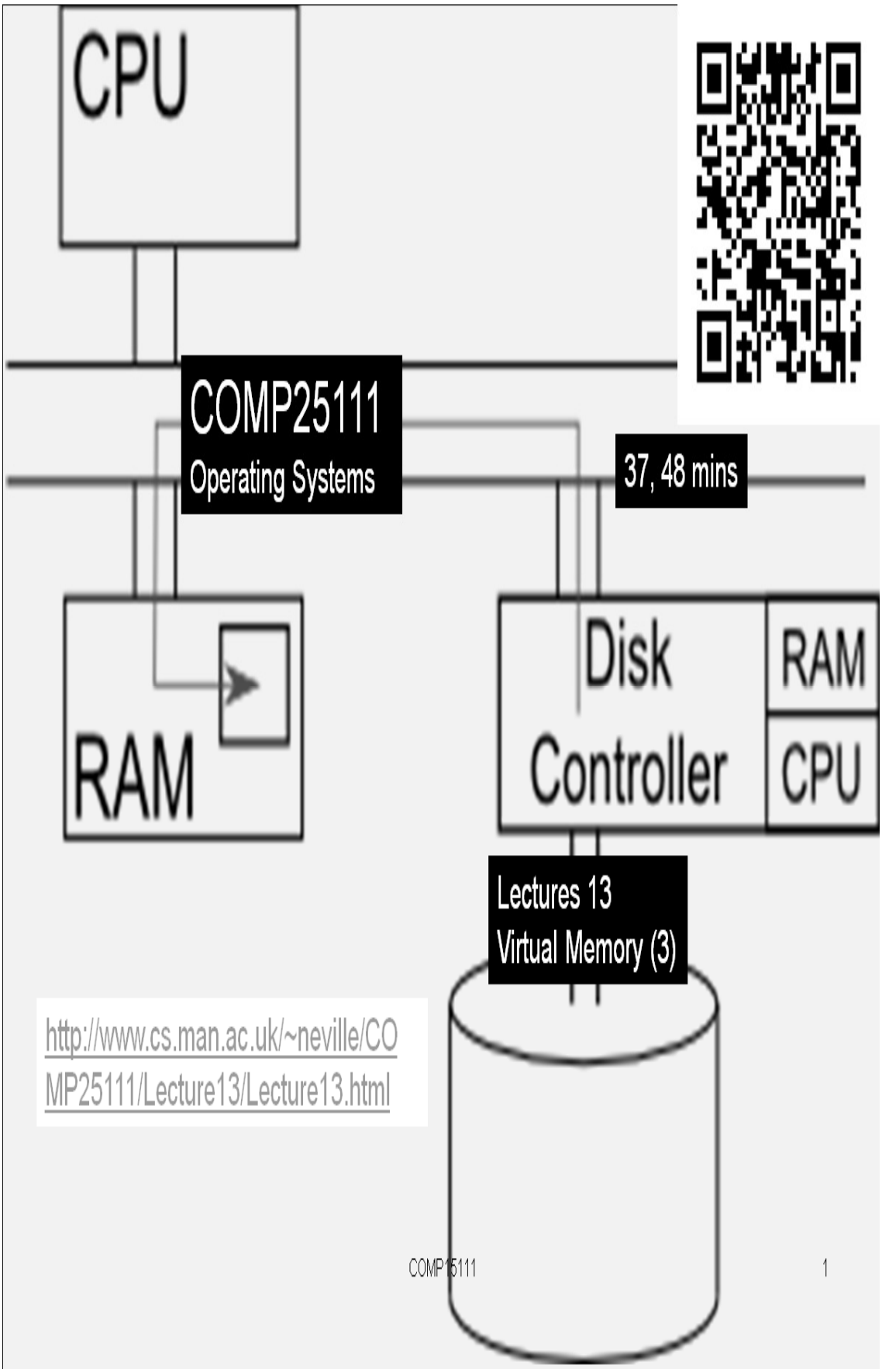
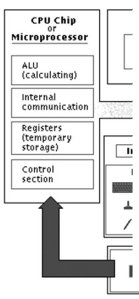
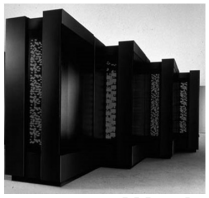

CPU
COMP25111
Operating Systems
37, 48 mins
RAM
Disk
Controller
RAM
CPU
Lectures 13
Virtual Memory (3)
http://www.cs.man.ac.uk/~neville/COMP25111/Lecture13/Lecture13.html

COMP25111
Operating Systems
Lectures 13: Virtual Memory (3)



Dr Richard Neville
r.neville@manchester.ac.uk



Room: G12 Kilburn Building, Bottom floor

Week 8

NOTE: The **up-to-date** version of this lecture is kept on the associated web site – available [on-line] @ Blackboard select: COMP15111 Introduction to Computer Systems **www.manchester.ac.uk/portal**

http://www.cs.man.ac.uk/~neville/COMP25111/Lecture13/Lecture13.html

Where to find this Lecture 13 of the COMP25111 course?

First Go to Blackboard 9; then select: COMP25111 Operating Systems

Then select: Week 8

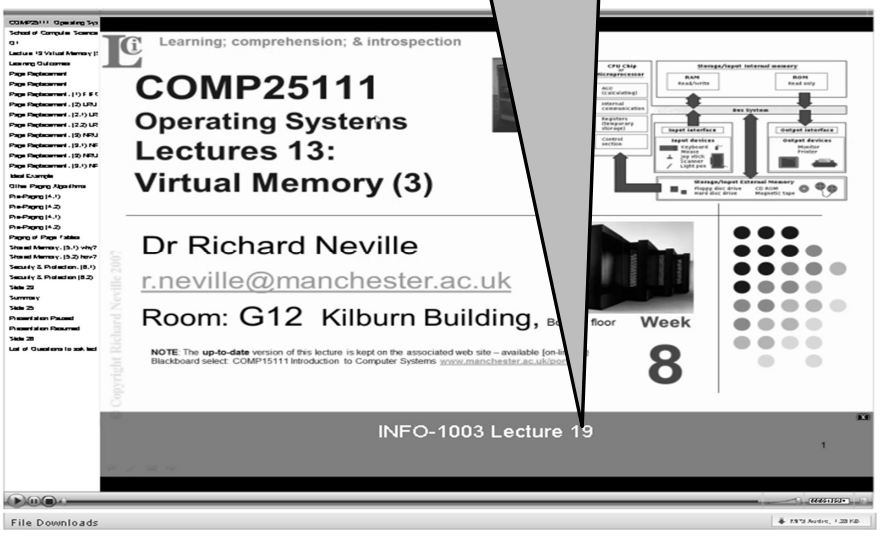
This topic provides...
12: Memory management 3 (Virtual Memory (2)) by RN;
13: Memory management 4 (Virtual Memory (3)) by RN.

Then select: Lecture 13 Information

Then select: Real Time Video of Lecture13

Then select:

Do not worry about the number "19" – keep watching the lecture and you will see it is actually"13"!!!



Students Copy

Short Exam Questions **Q1**

1. Question

   State the **three issues** Segmented Virtual Memory support with respect to the computer system as a whole.

   ANSWER(S):

   1)
   2)
   3)

   Answer(s):

- NOTE: In the exam approximately 2 question are taken from the topics (and program examples) coved in each lecture.

Now on Blackboard:

Lab3 Covers: Lab3 Exercise MSWord document

Lab 3 Folder Contains:

1. MMUSim.java; and
2. skeleton MMU.java [Java files]

Lab 3 Hints Folder: [#H1] and [#H2]

Lecture 13
Virtual Memory (3)

Efficient Use of Main Memory

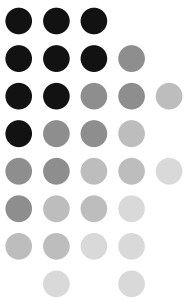
Towards Modern Operating Systems

Introduction to:

- Page Replacement Policies

Reading: MOS;

Reference: Modern Operating Systems (MOS) Andrew Tanenbaum,



Learning Outcomes

D-words C-words

Be able to

1) Discuss the concept of 'Page Replacement'
2) Differentiate between First in First Out (FIFO) and Least Recently Used (LRU)
3) Explain Not Recently Used (NRU)
4) Describe Pre-Paging
5) State why and how Shared Memory is used
6) Assess Security & Protection issues related to memory

FOOTNOTE2: D-words: direction words. C-Words: content words.
Ref.: Michael J. Wallace (1980, 2004)
Study Skills in English, ISBN 9780521537520.
D-Words also aligned to; Ref.: Taxonomy of Educational Objectives: The Classification of Educational Goals; pp. 201–207; B. S. Bloom (Ed.) Susan Fauer Company, Inc. 1956.

Footnote

1: Discuss: aligned to describe, paraphrase, discuss & give example.
2: Differentiate: aligned to analyze, categorize, compare and contrast.
3: Explain: aligned to describe; discuss; give examples.
4: Describe: aligned to paraphrase, discuss & give example [pictorial or diagrammatic].
5: State: aligned to define, identify.
6: Assess: aligned to evaluate and interpret.

Reference: Bloom, B.S., *Taxonomy of Educational Objectives.* Handbook I: The Cognitive Domain. 1956: New York: David McKay Co Inc.

Page Replacement

(p)	(i)
...	...
07	00
06	XX
05	03
04	02
03	XX
02	01
01	XX
00	XX

- Recall the MMU operation:

1. Given a Virtual Address;
2. Extract Virtual page number;
3. Look at corresponding page table entry;
4. If page is in real memory, access it;
5. else access the disk to read wanted page and find a place to put it in real memory. This will probably involve finding a page to reject to disk.

Page Replacement

- So which page to reject?
  - This can have a dramatic effect on performance.
- If we could predict the page which would be accessed again furthest in the future (maybe never) this would be the ideal one to pick!
- But (in any real program with dynamic behaviour) this is impossible.

Page Replacement: (1) FIFO

- First in First Out (FIFO)
  - Identify the oldest page in real memory and get rid of that;
    - they can be tagged with a **sequence number**.
  - But just because it's old doesn't mean that it isn't wanted!
    - It may contain a critical bit of code which is continually in use.
  - Can be improved using **second chance** algorithm – see MOS4.4.

Page Replacement: (2) LRU

A few additions to notes are required.

- Least Recently Used (LRU)
  -

Inverse of LRU is:

- If something has been used recently then it probably will be again. And the converse.
  - Simplest implementation is to associate a timestamp counter (may need 64 bits) with every page, updated whenever the page is accessed.

timestamp is the naming convention used in the Lab exercise.

Footnote 1: Temporal locality: The concept that, likelihood of referencing a resource is higher if a resource near it was just referenced.

Page Replacement: (2.1) LRU Continued

LRU Page Replacement timestamp

- But, if needed on every access, cannot be done efficiently in memory.
- Instead add **counter** to each TLB entry

Translation Lookaside Buffer

- But this added **counter** is:
  - Expensive hardware and can triple size of each page table entry.

Reference Lecture 13 Self-study NOTES: Translation Lookaside Buffer notes.

Learning; comprehension; & introspection

Page Replacement: (2.2) LRU Example

Physical Memory

Analogues to: **Virtual Memory**

1) Real memory has space for 3 page frames only
2) 5 pages (say 1,2,3,4,5) initially on disk [not in memory]
3) Requests in the order: 1 2 3 4 1 2 4 3 5 4

Requested by 'Execution Order' say…

Requests order →

	1	2	3	4	1	2	4	3	5	4
3 page frames only	1	2	3	4	1	2	4	3	5	4
		1	2	3	4	1	2	4	3	5
			1	2	3	4	1	2	4	3
	M	M	M	M	M	M	H	M	M	H

Requests order

Time Since used

M=miss

H = hit

LRU

Learning; comprehension; & introspection

Page Replacement: (3) NRU Not Recently Used

- Recently used pages kept in memory; this infers not recently used pages replaced.
- The R and M bits in a page table are used:
  - The R bit set when referenced.
  - The M bit set when modified (written to).
- How it works:
  1. At fixed intervals, the clock interrupt triggers and clears the referenced bit (R = 0) of all the pages.
  2. Referenced bit marks pages referenced in interval.
  3. So during interval if page referenced R=1 then it is **used**.
  4. If not R=0 then it is **NOT used**.
     - and at the end of interval is a candidate for replacement.
- Note: NRU is a crude approximation to LRU.

R bit page 0: **0**0001111
R bit page 1: **0**0000000

Reference: R bit is the referenced bit (read or written) – set if the page frame is referenced.
Other bits in the page table are: Modified, Present/absent + set of protection bits (R,W,X).

Page Replacement: (3.1) NRU Not Recently Used - Continued

- When replacing a page, the operating system divides the pages into four classes:

Question: State what classes '0' and '3' represent?

Not Recently Used

Class	R bit (referenced)	M bit (modified)
0	0	0
1	0	1
2	1	0
3	1	1

Recently Used

- The NRU algorithm picks a random page from the lowest class for removal.
  - Algorithm implies that a referenced page is more important than a modified page.

Idealised page replacement algorithm

Ideal Example

Belady's optimal replacement algorithm

Order: 1 2 3 4 1 2 4 3 5 4

Look forward to 'furthest future' use

3 page frames only

1	2	3	4	1	2	4	3	5	4
1	2	3	4	4	4	4	4	4	4
	1	2	2	2	2	2	2	5	5
		1	1	1	1	1	3	3	3
M	M	M	M	H	H	H	M	M	H

Requests order

M=miss

H = hit

S Middle E

Other Paging Algorithms

- There are a number of refinements to the LRU principle resulting in other algorithms.
- To emphasise, because any page fault causes a disk access, it takes time (***mS***) compared to ***nS*** instruction execution.
- Minimising page faults by using sophisticated replacement algorithms is worthwhile.

Pre-Paging (4.1)

- If a swapped out process is brought back into memory, we could let it reload its new pages one by one.
- However, it will be much more efficient if we keep track of the process' **working set**
  - i.e. group of recently used pages, and
- load these in one go before re-starting the process.

Footnote 1: Pre-paging: A technique whereby the operating system in a paging virtual memory multitasking environment loads all pages of a process's working set into memory before the process is restarted.

Learning; comprehension; & introspection

Pre-Paging (4.2)

- We may also be able to keep track of groups of pages that get used together when a process is running.
- Rather than just load one at a time we can load a **group** together.
- Obviously need to displace others, so need to be careful. But used in practice (Linux 2.4 + kernel).

Learning; comprehension; & introspection

Paging of Page Tables

- Page tables are just locations in memory which exist on pages.
- They can themselves be paged out to disk.
- However, when a process is running, its page tables must be in memory.
- In practice, only those parts of the table which hold translations for the page tables!

Shared Memory: (5.1) why?

A few additions to notes are required.

- In some cases it is necessary for processes to share memory:
  1) [ ]
  2) Shared data space (e.g. Unix pipes); or
  3) Shared Library Code (dll's, i.e. dynamic link libraries).
- If each process sees only its own version of virtual memory space, how can this happen?

Shared Memory: (5.2) how?

- Page tables can be set up so that (possibly different) virtual addresses in multiple processes translate to the same real page.
- This would normally be organised by the OS.
- [But] Allowing user processes to alter their own page tables would clearly be a major security issue.

Security & Protection: (6.1) Reason

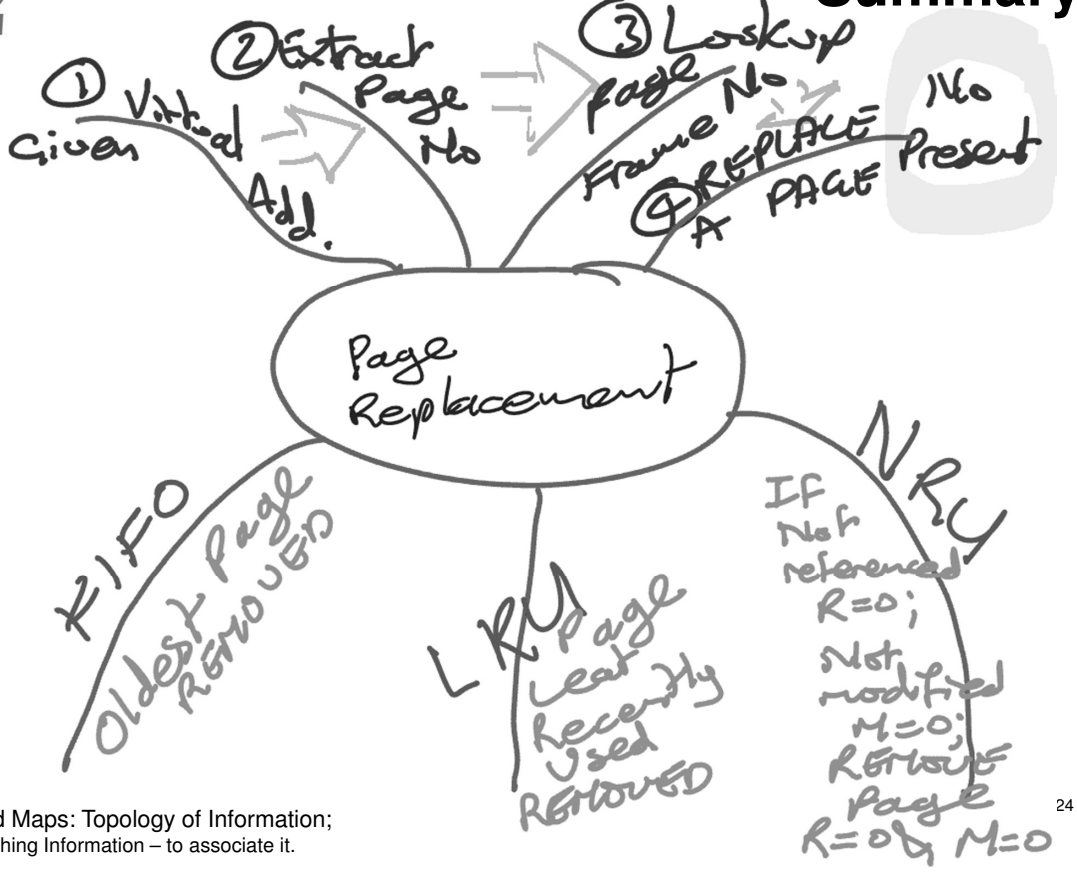
- We have seen that a page table entry can contain permission information (R,W,X).
- This can be used to control access;
- for example:
  1) Code can be marked 'read only'; (R,W,X)=(1,0,0)
  2) Page tables can be marked 'read only'; or (R,W,X)=(1,0,0)
  3) Data can be 'read/write' but **not** 'execute'. (R,W,X)=(1,1,0)
     - avoid accidental or malicious execution of data.

Security & Protection (6.2)

- But pages are there to help manage the address space.
- They are not natural units of protection. E.g.
  1) Code Area;
  2) Data Area; or
  3) Stack Area.
- Some (but not all) systems instead have an additional level of virtual memory.

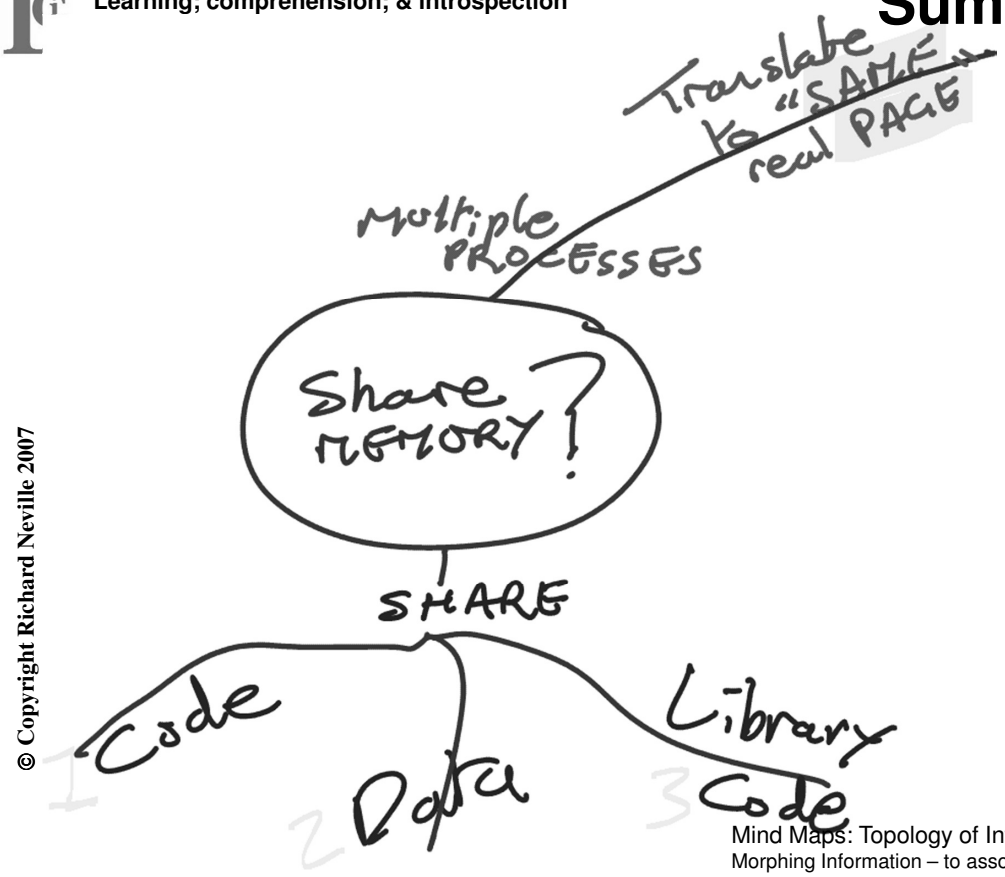
END

Summary

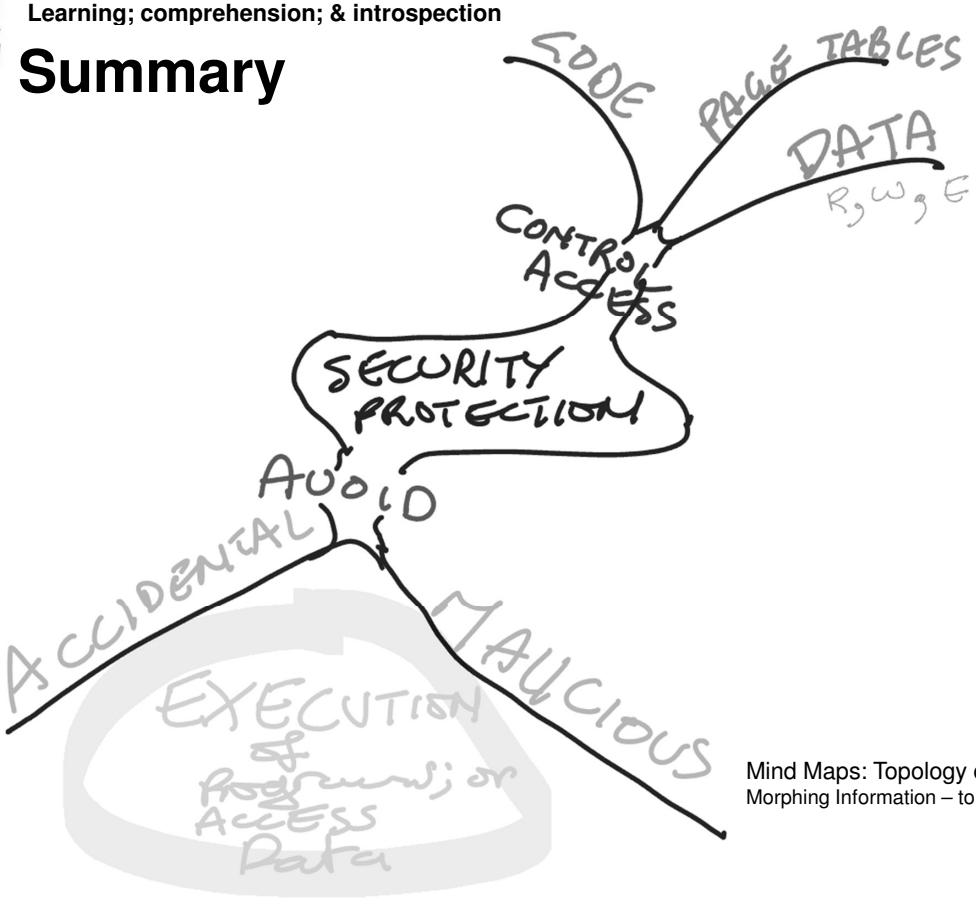


Mind Maps: Topology of Information;
Morphing Information – to associate it.

Summary



Summary



List of Questions to ask lecturer

- Before the 9a.m. start lecture the lecturer will be half an hour early and you can ask [any and all] questions in that half hour; before the lecture:

1.

2.

3.

4.

5.

Getting ready for next week
Do next week's Q3's NOW

- Once you have re-read the lecture notes; and listened to the audio recording [while stepping through the PPT] of the lecture again:
- Please have a think about next week's Q3's
  - on the next page
- If you try to answer the Q3's now you will be in a much better position to recall the information.
- Once you have done this, transfer your answers to next weeks "Student [OWN answers] version" at the start of next weeks lecture.
  - YES this implies bringing the last weeks lecture notes to the next lecture …

Learning; comprehension; & introspection

Now on Blackboard:
Lab3 Covers: Lab3 Exercise MSWord document
Lab 3 Folder Contains:
MMUSim.java; and
skeleton MMU.java [Java files]
Lab 3 Hints Folder: [#H1] and [#H2]

THIS week's

Short Exam Questions **Q3**

1. Question
   State the **three issues** Segmented Virtual Memory support with respect to the computer system as a whole.

Answer(s):

2. Question
   Give two reasons why 'segments' support the management of the execution of multiple processes in an operating system :

Answer(s):

3. Question
   Give the names of the two fields the segment address is split into.

Answer(s):

- NOTE: In the exam approximately 2 question are taken from the topics (and program examples) coved in each lecture

Learning; comprehension; & introspection

Next lecture [week's]

Short Exam Questions **Q3**

1. Question
   Name three page replacement policies?

Answer(s):

2. Question
   What is meant by the term 'working set;' with respect to process memory pages?

Answer(s):

3. Question
   What methodology can minimise page faults by the use of sophisticated replacement algorithms?

Answer(s):

- NOTE: In the exam approximately 2 question are taken from the topics (and program examples) coved in each lecture

GLOSSARY

- Using the on-line resources and any other resources compile a glossary of the terms below:
- Page replacement →
- Virtual address →
- Page table →
- FIFO →
- LRU →
- Timestamp counter →
- Page frames →
- Page number →
- Working set →
- Translations →
- User code →
- Shared data space →
- Shared library code →
- Permission information →

Learning Resources 1

- **Descriptions [Theory] (in text books)**
- Remember the key issues, highlighted in GREEN, are the concepts to look for in any book:
  - Section on page replacement algorithm, first-in-first-out replacement algorithm, least recently used replacement algorithm in chapter 6 the Central processor unit operation in: Chalk BS, Carter AT, Hind RW (2004) Computer Organisation and Architecture: An introduction 2nd Edition, Palgrave, ISBN 1-4039-0164-3 .
  - Section on page replacement algorithm, FIFO, LRU, NRU, Pre-paging, Security, protection [memory] – in chapter 4 the Computer Systems Organization; & Parallel Computer Architectures in: Modern Operating Systems, 2/E, Andrew S. Tanenbaum, Vrije University, Amsterdam, The Netherlands, Publisher: Prentice.

- **Web resources:**
  - **Virtual Memory; available [on-line] @ http://courses.cs.vt.edu/~csonline/OS/Lessons/VirtualMemory/index.html**
  - **Cache Simulator Lab Session 1; available [on-line] @ http://myweb.lsbu.ac.uk/~chalkbs/research/CacheSimDescription.htm**
  - **Virtual Memory Page Replacement Algorithms [good set of references at the end]; available [on-line] @ http://people.msoe.edu/~mccrawt/resume/papers/CS384/mccrawt_cs384_virtual.pdf**
  - **MOS Free e-book [Low resolution (Not high quality graphics or printing – but readable)]:**
  - **Modern Operating Systems (MOS) 2nd Edition Andrew Tanenbaum, available**
  - **[on-line] @: http://www.freebookzone.com/fetch.php?bkcls=os_thry&bkidx=35**
  - **Memory *Replacement* Policies ; available [on-line] @ *www.cs.uiuc.edu/class/fa08/cs241/lectures/24-Memory.ppt***

Questions

Introduction to Questions:

The set of questions are based on lecture 13.

Answer Sheet will be given later in year and will contain the answers to these questions.

- Remember to find detailed and comprehensive answer you should [also] reference associated text books in the library.
- A reasonable starting place for associated book titles are:

1) This units 'module guide'; given to you in RN's first lecture – or on the web [Blackboard];
2) Those books mentioned in 'Background Reading;'
3) Those books [and web resources] mentioned in Learning Resources.

Long [& Short] Exam Questions

Questions

1. Question:

- Draw up a table that lists:
- 1) [Page] Replacement policy name; &
- 2) Brief description of how the policy works.

1. Answer:

Policy name	Description of how the policy works

Answer(s):

Learning; comprehension; & introspection
Long [& Short] Exam Questions
Questions

2. Question

State three reasons why it is necessary for processes to share memory.

2. Answer

Answer(s):

3. Question

In the context of permission information: like read (R), write (W), & execute (X).

What can the RWX permission information control access to; with respect to security and protection?

3. Answer

Answer(s):

4. Question

On a paged machine with 3 pages frames available for it, a particular process makes accesses to the following pages in the order given:

0, 7, 3, 1, 2, 3, 1, 7, 3

Show the contents of the 3 page frames and the cumulative total number of page faults after each memory access assuming that an LRU page replacement algorithm is in use and that the page frames are initially empty. The type of diagram you should draw up is depicted in figure below:

Access:	0	7	3	1	2	3	1	7	3		
Most recent:	X	X	X	X	X	X	X	X	X	)	Contents of
Second most :	X	X	X	X	X	X	X	X	X	)	page frames
Third most:	X	X	X	X	X	X	X	X	X	)	in memory
Total PFs:	X	X	X	X	X	X	X	X	X		

Typical diagram showing 3 page frames and the cumulative total number of page faults.

4. Answer

For 0, 7, 3, 1, 2, 3, 1, 7, 3:

Answer(s):

5. What is a page replacement algorithm?

Answer(s):

Revision Exercises

- Scan read Lecture 13's Questions.
  - Answer Lecture 13's Questions
    - Particularly those questions you had difficulties with when you first tried them.

Background Reading

- [1] Computer Organisation and Architecture: An introduction 2nd Edition, Palgrave, ISBN 1-4039-0164-3 chapter 6 the Central processor unit operation in: Chalk BS, Carter AT, Hind RW (2004).
  - Section on page replacement algorithm, first-in-first-out replacement algorithm, least recently used replacement algorithm.
- [2] Modern Operating Systems, 2/E, Andrew S. Tanenbaum, Vrije University, Amsterdam, The Netherlands, Publisher: Prentice
  - Section on page replacement algorithm, FIFO, LRU, NRU, Pre-paging, Security, protection [memory].
- {On Web} **4.4 First-In, First-Out (FIFO),** Ref. **Page replacement in operating system memory management, by Heikki Paajanen,** available on-line from [URL]: https://jyx.jyu.fi/dspace/bitstream/handle/123456789/12528/URN_NBN_fi_jyu-2007775.pdf, (last visited [accessed]: 21-7-2011).
- {On Web} **Memory Management: Virtual Memory,** Ref. Understanding Operating Systems, Fifth Edition available on-line from [URL]: http://www.johnrouda.com/class/PDF/CPT%20257/01600_IM_ch03.pdf, (last visited [accessed]: 21-7-2011).
- {On Web} **Part 2: Paging Algorithms and Implementation Issues**, Ref. Chapter 4: Memory Management, on-line from [URL]: http://www.cs.pitt.edu/~mosse/cs1550/Slides/amer-memory2.pdf, (last visited [accessed]: 25-7-2011).
- {On Web} **4.4 PAGE REPLACEMENT ALGORITHMS,** Ref. MEMORY MANAGEMENT, CHAP. 4 , on-line from [URL]: http://www.lira.dist.unige.it/teaching/OS/files/sample-4.pdf, (last visited [accessed]: 25-7-2011).
- {On Web} **,** Ref. , on-line from [URL]: , (last visited [accessed]: 25-7-2011).