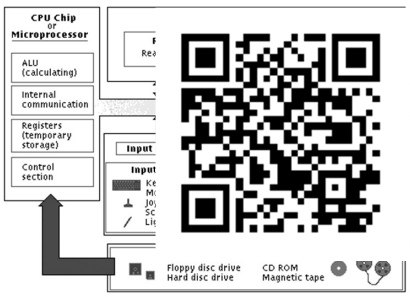
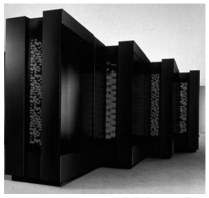
Learning; comprehension; & introspection

COMP25111
Operating Systems
Lectures 12
Virtual Memory (2)
Segmented Virtual Memory



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: COMP25111 Introduction to Computer Systems **www.manchester.ac.uk/portal**

https://stream.manchester.ac.uk/Play.aspx?VideoId=9079

Learning; comprehension; & introspection

Where to find this Lecture 12 of the COMP25111 course?

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

Then select: Week 8

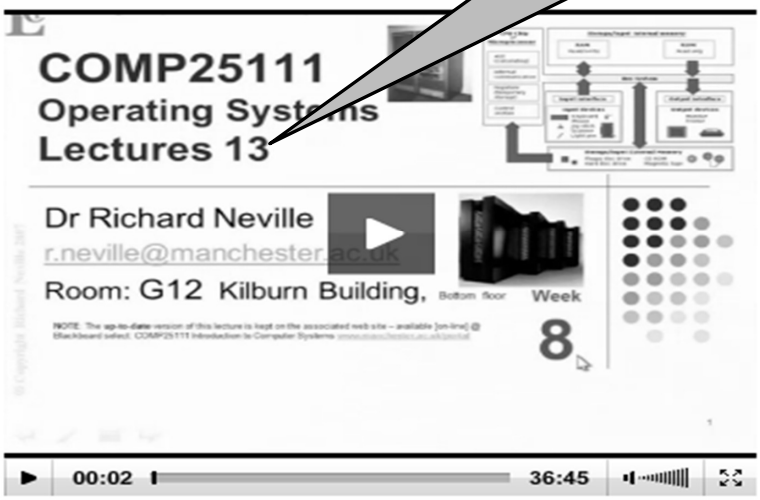
This topic provides...
12: Memory management 3 (Virtual Memory (2)) by RN;
13: Memory management 4 (Virtual Memory (3)) by RN.

Then select: Lecture 12 Information

Then select: Real Time Video Lecture12

Then select:

Do not worry about the number "13" – keep watching the lecture and you will see it magically changes to "12"!!!



THIS week's Short Exam Questions Q1

1. Question

   Differentiate between the write-through strategy and the write-back strategy.

   ANSWER(S):

Answer(s):

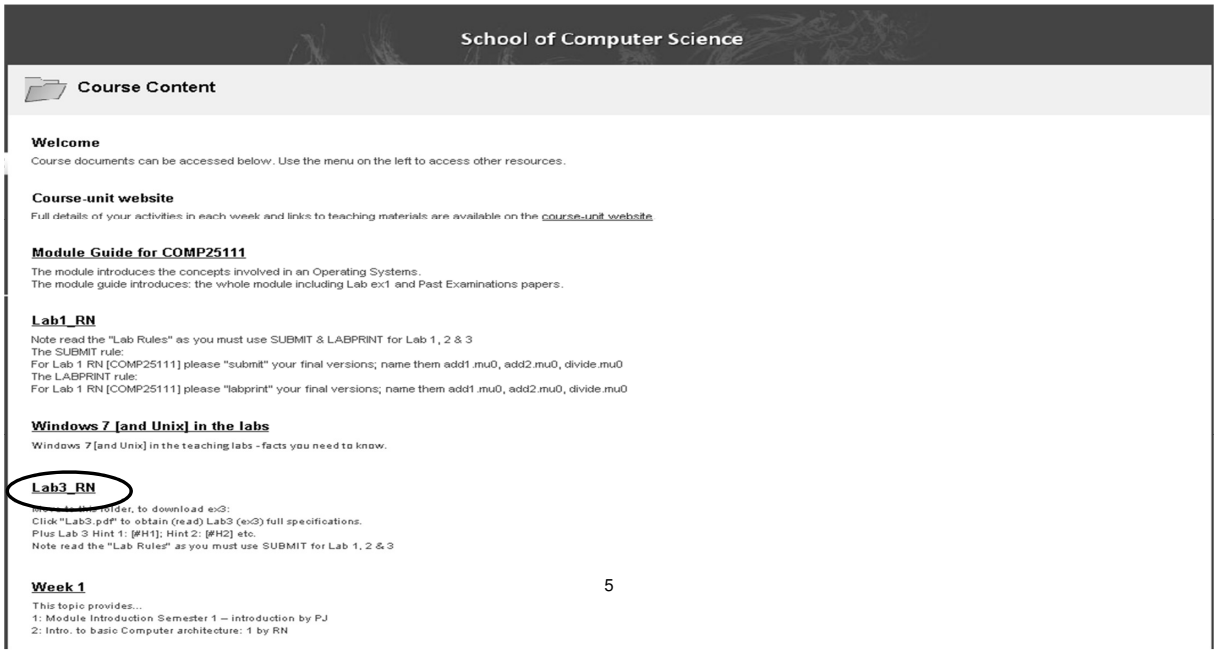
- NOTE: In the exam approximately 2 question are taken from the topics (and program examples) coved in each lecture.

Lab 3 RN

- If you are attending RNs Lectures L12 etc;
- You will be expected to undertake:
- Lab 3 RN:
- Time of lab:
  - Sem 1 A w3+ Lab LF31 Fri 09:00 - 11:00 F
  - Sem 1 A w3+ Lab LF31 Fri 11:00 - 13:00 I
  - Sem 1 A w3+ Lab G23 Mon 13:00 - 15:00 G

Lab 3 RN, cont. 1.

- All the information about the lab is available on Blackboard 9;
- Web Site: **Blackboard select:**
  COMP25111 Operating Systems
  www.manchester.ac.uk/portal



Lab 3 RN, cont. 2.

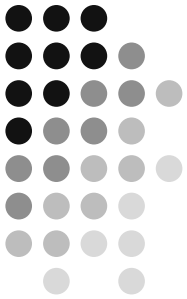
- Where to Find NetBeans on the lab machines:
  1) Boot into Windows 7;
  2) Click “Start;”
  3) Click “All programs;”
  4) Click the “NetBeans” folder [icon];
  5) Click the “NetBeans IDE 7.0.1” icon.

Lecture 12
Virtual Memory (2)

"The variably-sized block concept"

Segmented Virtual Memory

Self-study NOTES:
Combining Segmentation and Paging



Learning Outcomes

D-words

C-words

Be able to

1) Explain what is meant by segmented virtual memory
2) Apply first-fit and best-fit algorithms for *loading segments into memory*
3) Describe how segmented and paged virtual memory systems can be used
4) Explain how a translation lookaside buffer works (Self-study NOTES)

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: Explain: aligned to describe; discuss; give examples.
2: Apply: aligned to carry out, demonstrate and illustrate.
3: Describe: aligned to paraphrase, discuss & give example [pictorial or diagrammatic].

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

The Audio recording for:

Self-study NOTES: Using Paging and Segmentation Together;

Lecture 13 is on:

Blackboard 9

Directory:

Directory For: The audio for, Lecture 13; Self-study NOTES: Using Paging and Segmentation Together

Audio File Name:

L13SelfStudySegAndPagedMem.zip

Learning; comprehension; & introspection

Segmented Virtual Memory

A few additions to notes are required.

- In a paged virtual memory system, the pages are all the same size.
- In a segmented virtual memory system, the segments are of variable size.
- Segmentation is less about mapping a larger virtual address space onto a smaller physical memory (the purpose of paging), and more about **supporting** the computer system as a whole [three main issues]:

1)

2) **System software**; and

3) Programmers in managing processes in a **multitasking** operating system.

Re. **Learning Resources 14**, available at the end of the lecture notes.

Attributes of Segments

- A segment is a variably-sized block of virtual address space.
- Each segment can be viewed as an address space in its own right:
  - Starting from the address 0;
    - This simplifies the management of compiling and loading programs.
- Each segment has attributes associated with it:
  1) **Access rights** determine which programs can use the segment (particular users or the operating system); &
  2) **Usage rights** determine what operations can be performed on the memory (e.g. read-only, read-execute).

Segments

- Segments are a way of dividing [up] the virtual address space to support the management of the execution of multiple processes in an operating system, to:
  1) ensure that processes do not interfere with one another;
  2) ensure that operating system has control of the computer; and
  3) will prevent programs written by users taking over the computer.

Segment Address

A few additions to notes are required.

- The address is divided into two fields:
  1)
  2) Offset.
- The number of bits for the segment number is determined by the architecture of the processor.

Where to find this Lecture 12 Segmented Virtual Memory 1 & 2 Tutorial one of the Cam Coder Tutorials for the COMP25111 course?

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

Then select: Week 7

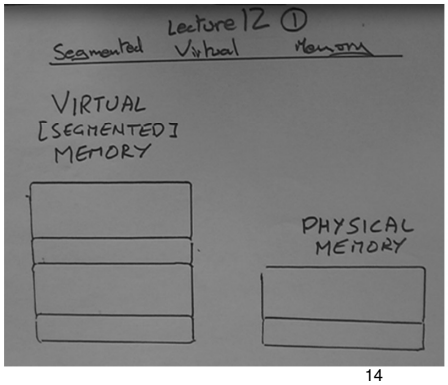
This topic provides...
10: Memory management 1 (Introduction to basics) by RN;
11: Memory management 2 (Virtual Memory (1)) by RN;

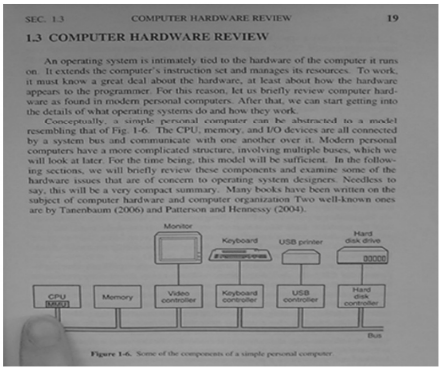
Then select: Lecture 11 Information and then: VIDEOS

Then select: Cam Coder Tutorial Lecture12 Segmented Virtual Memory 1

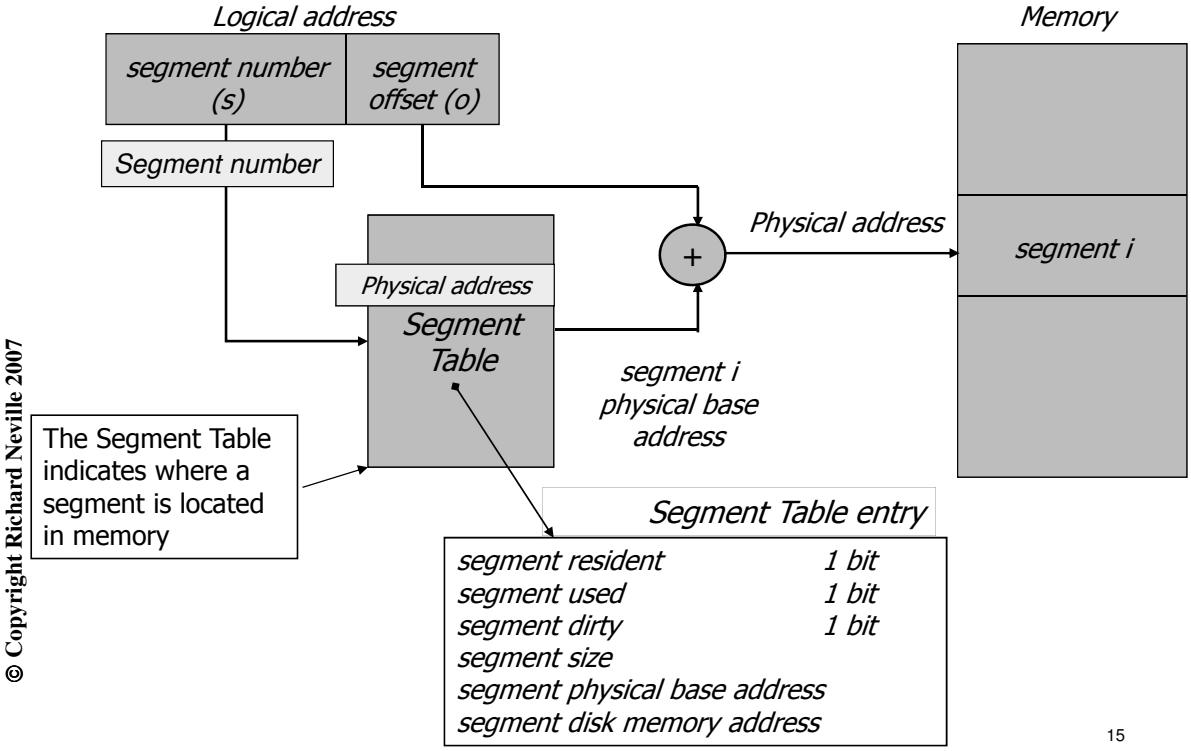
Then select: Cam Coder Tutorial Lecture12 Segmented Virtual Memory 2

Then select:





Segmented Virtual Address Mapping



Plus other permission information (R,W,X).

Segmentation process

or description of how a 'segment' is loaded.

- This works in almost exactly the same way as paged virtual memory.
- 1) The processor generates a logical address.
- 2) The segment number field is used by the MMU to look to see whether the segment is in memory or not, then:

3) If it is in memory, a physical address is computed by adding the base[1] address of the segment to the offset;

- this is used as a physical address to memory;

4) If it is not in memory, the transfer is aborted (segment fault) and the operating system will load the segment from disk to memory.

Footnote

1: Base address = where it starts in physical memory.

Segment Fault

- When a reference is made to a segment that is not in memory a segment fault (like a page fault) occurs.
- A similar procedure to a page fault is followed (Lecture 11, 'Paged Virtual Memory Procedure') and the segment is loaded into memory.

Segmentation

- The operating system must manage the positioning of segments in memory.
- This is more complex than in a paged system, because the segments are of variable sizes.
- A decision needs to be made as to where to place an incoming segment in the physical memory.

Learning; comprehension; & introspection

Where to find this Lecture 12 External Fragmentation Tutorial one of the Cam Coder Tutorials for the COMP25111 course?

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

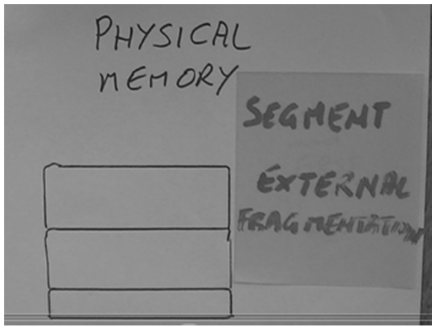
Then select: Week 7

This topic provides...
10: Memory management 1 (Introduction to basics) by RN;
11: Memory management 2 (Virtual Memory (1)) by RN;

Then select: Lecture 11 Information and then: VIDEOS

Then select: Cam Coder Tutorial Lecture12 External Fragmentation

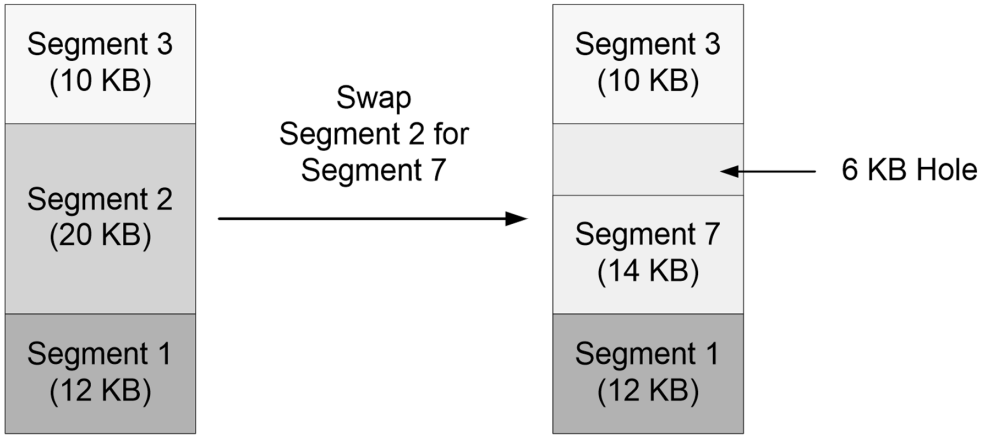
Then select:



Learning; comprehension; & introspection

Segmentation

External Fragmentation can occur, where memory space is wasted due to 'holes' in the physical memory



Where to find this Lecture 12 Avoiding External Fragmentation Tutorial one of the Cam Coder Tutorials for the COMP25111 course?

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

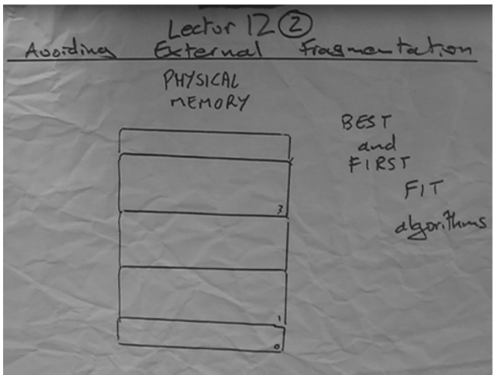
Then select: Week 7

This topic provides...
10: Memory management 1 (Introduction to basics) by RN;
11: Memory management 2 (Virtual Memory (1)) by RN;

Then select: Lecture 11 Information and then: VIDEOS

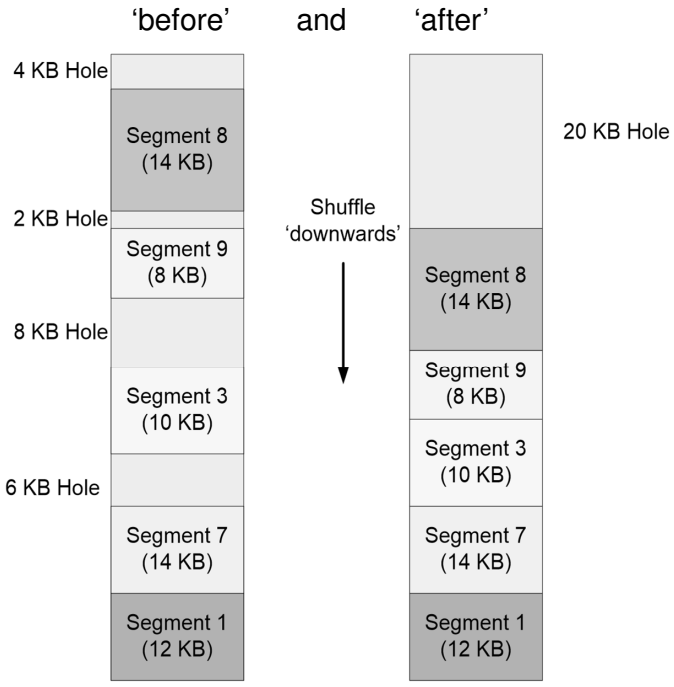
Then select: Cam Code Tutorial Lecture 12 Avoiding External Fragmentation

Then select:



Avoiding External Fragmentation

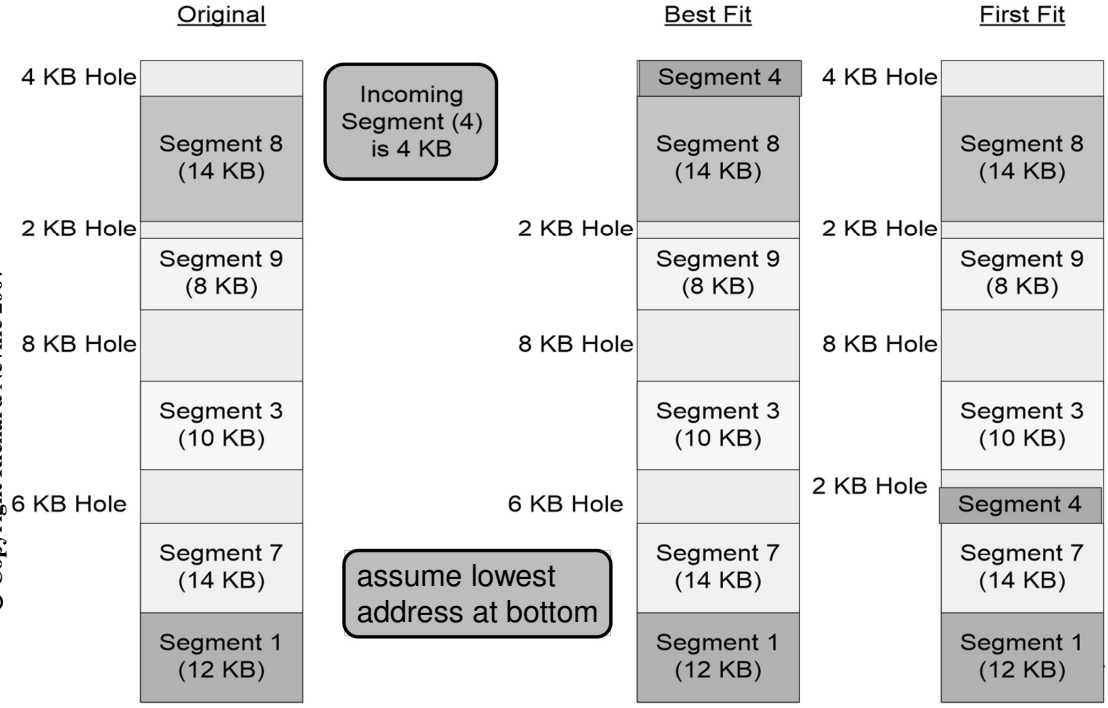
- Compact the memory by 'shuffling' segments in memory to fill the holes.
- But this requires extensive copying of data and is excessively time consuming.



Avoiding External Fragmentation

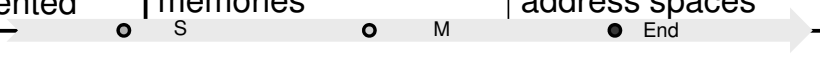
- Alternatively, an algorithm can be used to determine where to place the segment in a memory that has external fragmentation.
- The operating system maintains a list of the addresses and sizes of all the holes and can use algorithms like:
  1. Best Fit – scan the complete 'list of holes' and determine which best fits the segment;
     - Tends to produce a lot of small holes.
  2. First fit – scan the list until a [the first] hole is found that fits the segment.

External Fragmentation: Simple Example



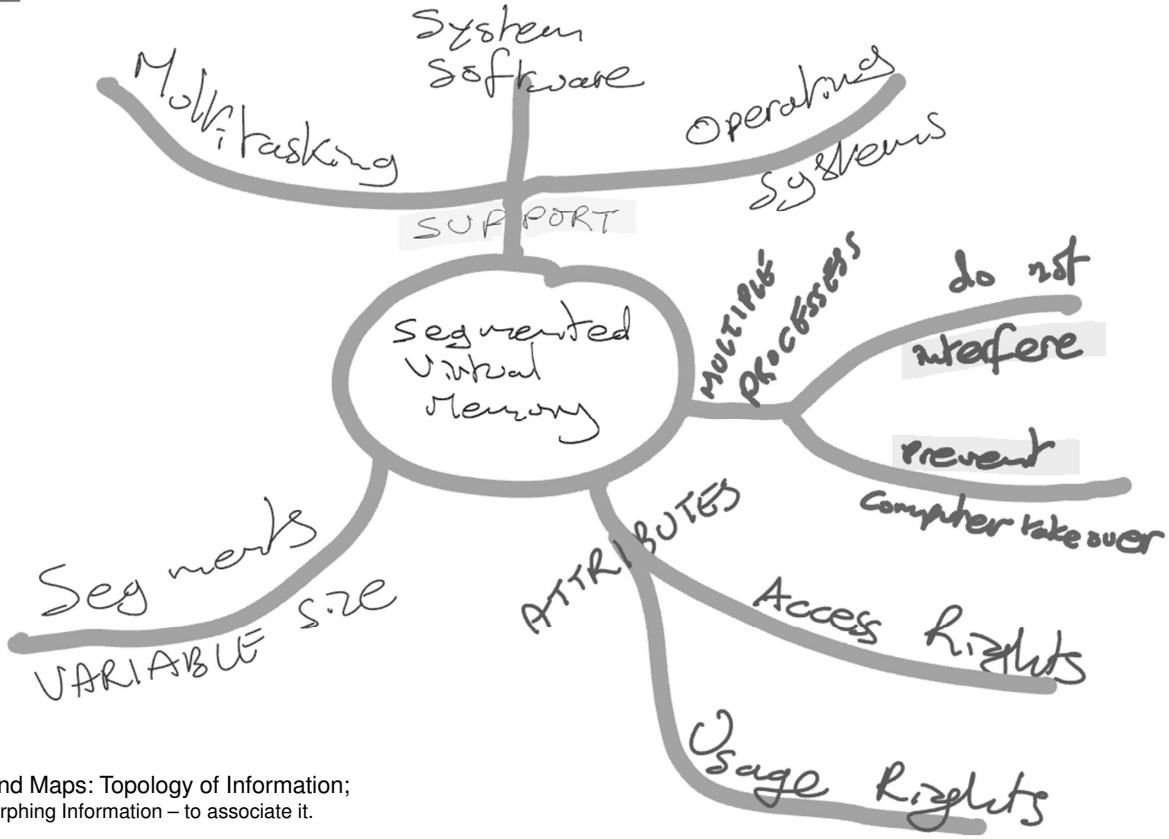
Paging versus Segmentation

Issues	Paging	Segmentation
Programmer Visible?	No	Yes
Block Replacement	Trivial (all blocks the same size)	Hard (must find unused portion of memory)
Memory Use	Internal Page Fragmentation	External Page Fragmentation
Efficient Disk Traffic?	Yes (adjust page size to balance access and transfer times)	Not always (small segments can be just a few bytes)
Why was the technique invented	To simulate large memories	To provide multiple address spaces



END

Summary

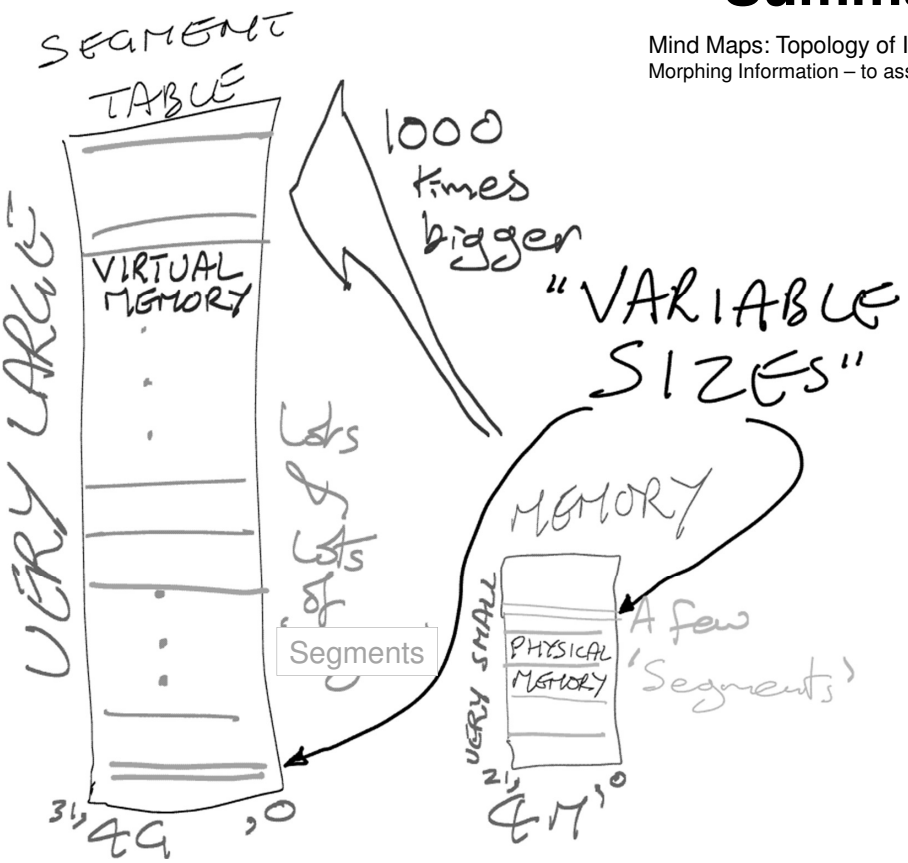


Mind Maps: Topology of Information;
Morphing Information – to associate it.

Learning; comprehension; & introspection

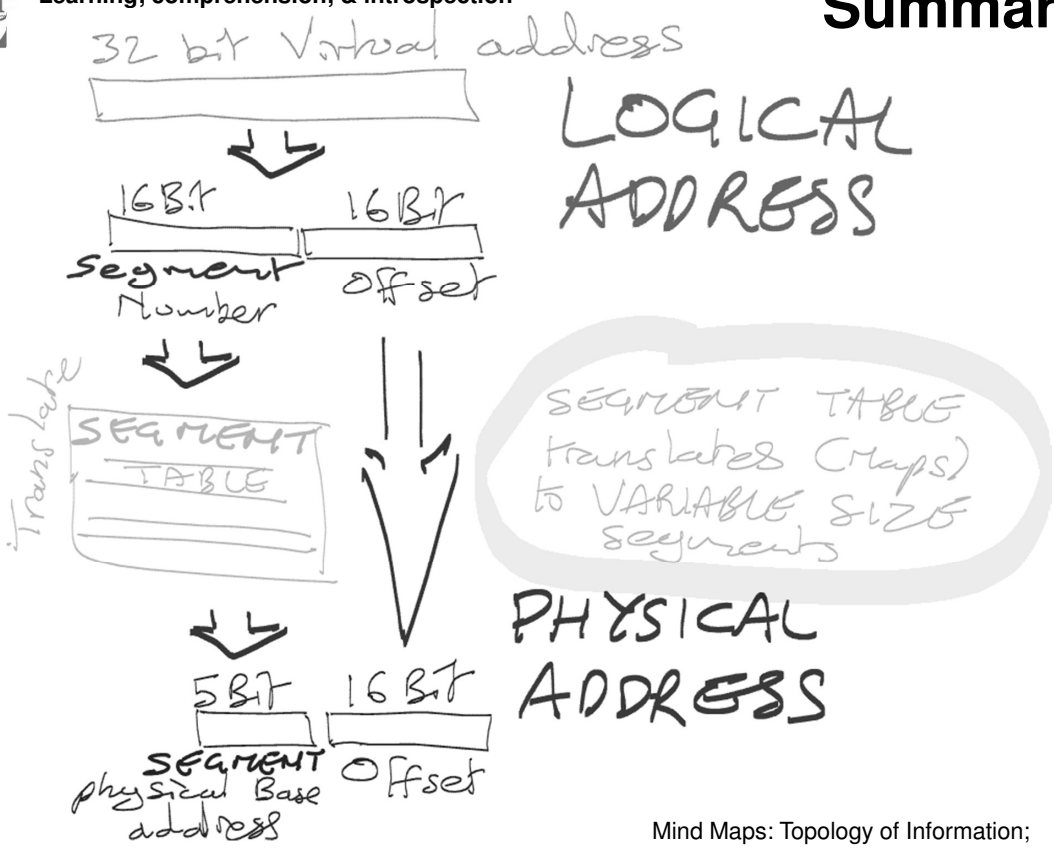
Summary

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



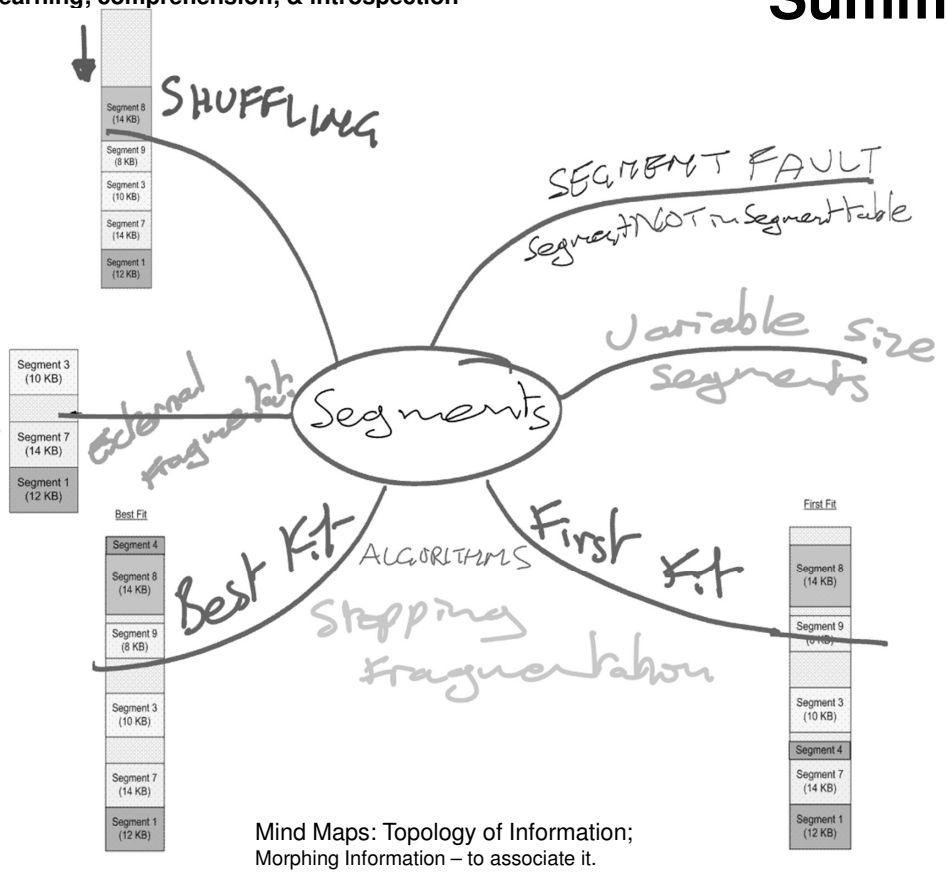
© Copyright Richard Neville 2007

Summary



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

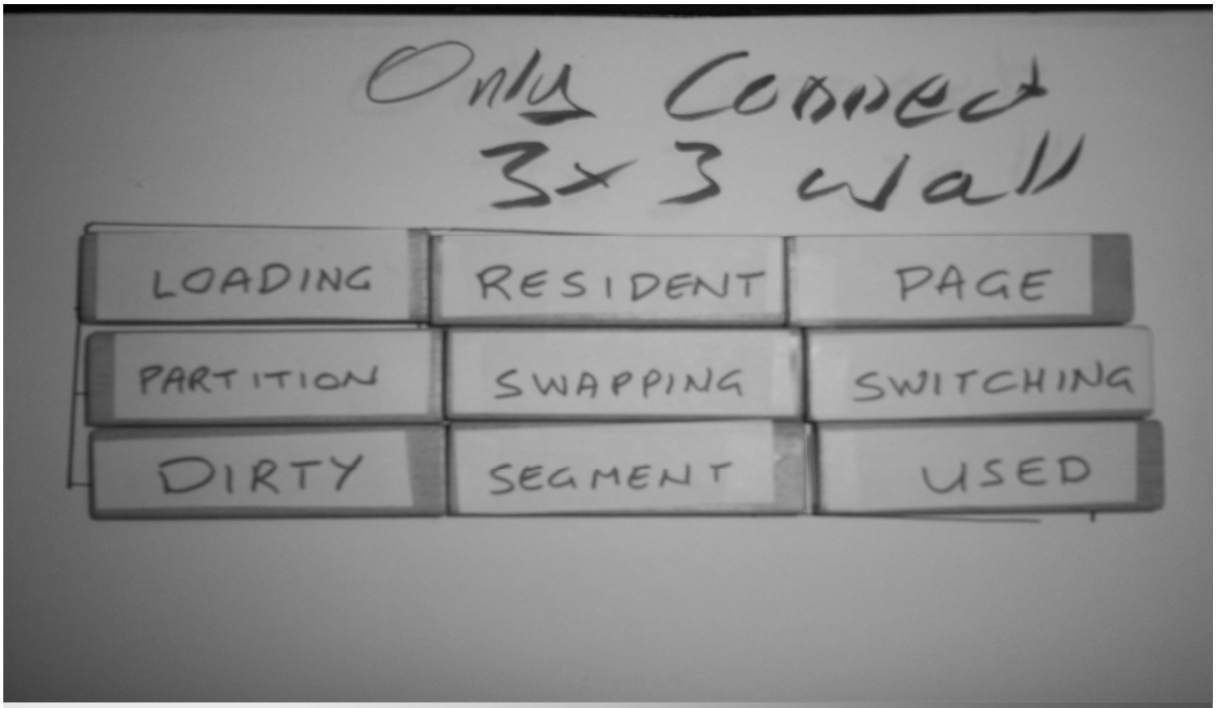
Summary



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


Only
Connect
Quiz


Only
Connect
The Rules, how to play
http://youtu.be/59qwTcAceTA



Quiz

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 …

THIS week's Short Exam Questions **Q3**

1. Question
   Differentiate between the write-through strategy and the write-back strategy.

   Answer(s):

2. Question
   What does the dirty bit indicate. State how it is utilised.

   Answer(s):

3. Question
   Discuss the differences between '*page fault*' and '*page demand*.'

   Answer(s):

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

SEE: **GLOSSARY & AUX. DATA for expanded answers.**

Next Lecture [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):

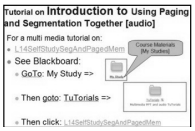
3. Question
   Give the names of the two fields the segment address is split into.

   Answer(s):

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

Self-study NOTES:
Using Paging and Segmentation Together

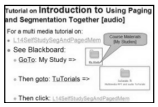
- Segmentation and paging are often used together.
- Virtual memory is divided into segments, each segment is divided into pages:
  1) Segments provide the multiple virtual address spaces that support operating systems and other system software; &
  2) Pages provide a mapping between a smaller physical memory and larger virtual memory;
     - Transferring pages from disk to memory.



Self-study NOTES:

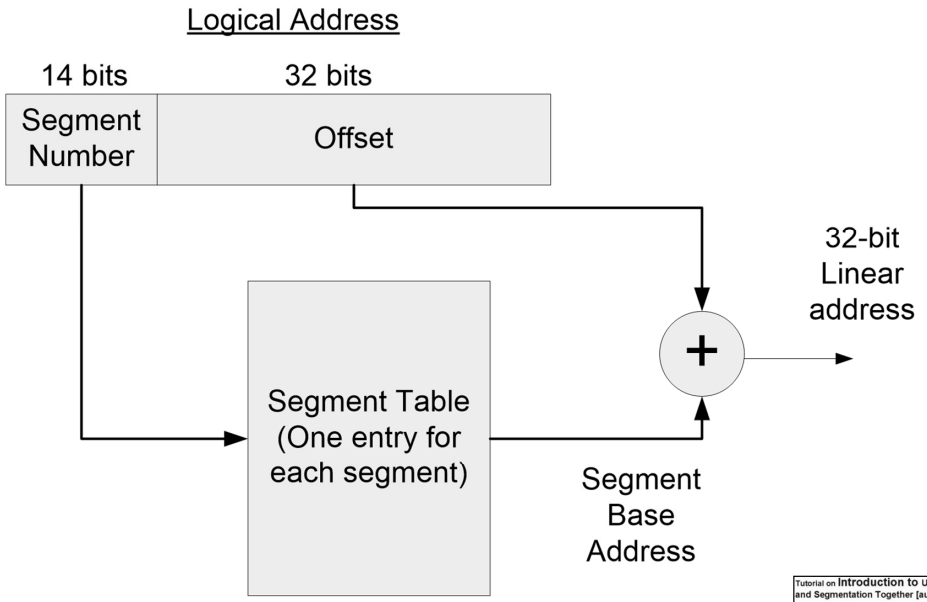
Segmented Paged Memory: An example

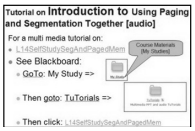
- This example is based on a version of the **Intel Pentium** processor;
  - Some of the details have been simplified.
- The processor generates a 46-bit virtual address
  - This consists of a 14-bit segment number and a 32-bit offset;
  - This means the virtual address space is 2^{46} (64 TB).
    - T = Terra = 2^{40}.
  - Divided into 2^{14} (16K) segments, each with an address space of 2^{32} (4MB).
    - The virtual address space of each segment is as large as the physical address space of the processor:
      - 32 address line externally.



Self-study NOTES:

Segmented Addressing

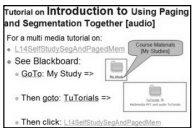




Self-study NOTES:
Paging

- Pages are divided in two levels:

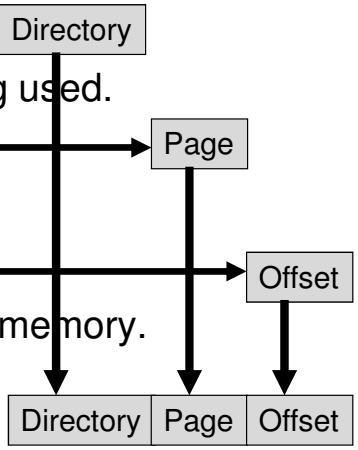
1) At the highest level there is a page directory:
   - This is a table with 1024 entries (2^{10}); &
   - Each directory has its own page table.
2) At the lowest level there is a page table:
   - This is a table with 1024 entries (2^{10})

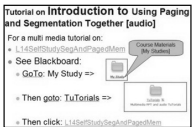


Self-study NOTES:
Paging

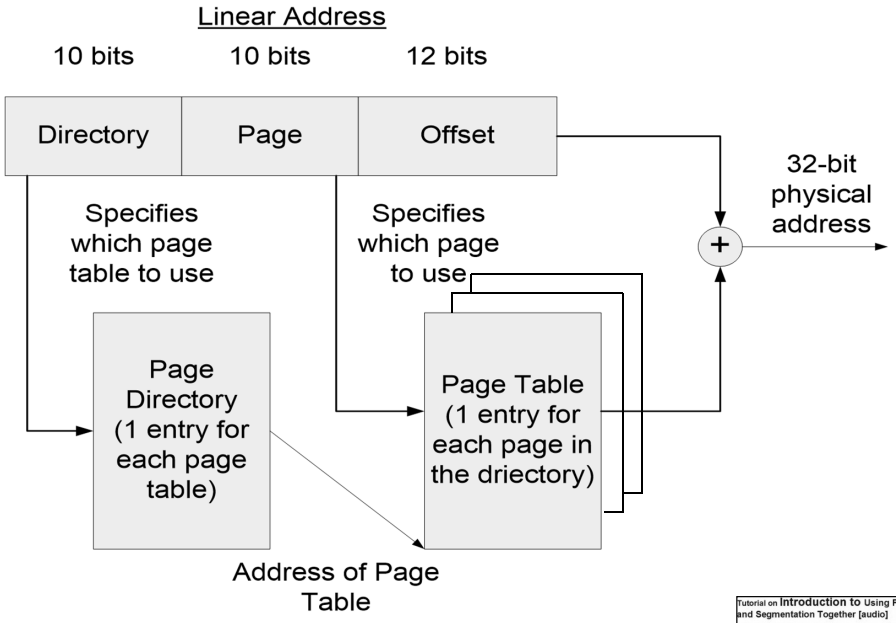
- The 32-bit linear address is broken into 3 fields:

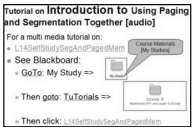
1) 10-bit page table field;
   - States which page table is being used.
2) 10-bit page field;
   - State which page is being used.
3) 12-bit offset;
   - To address the byte in physical memory.



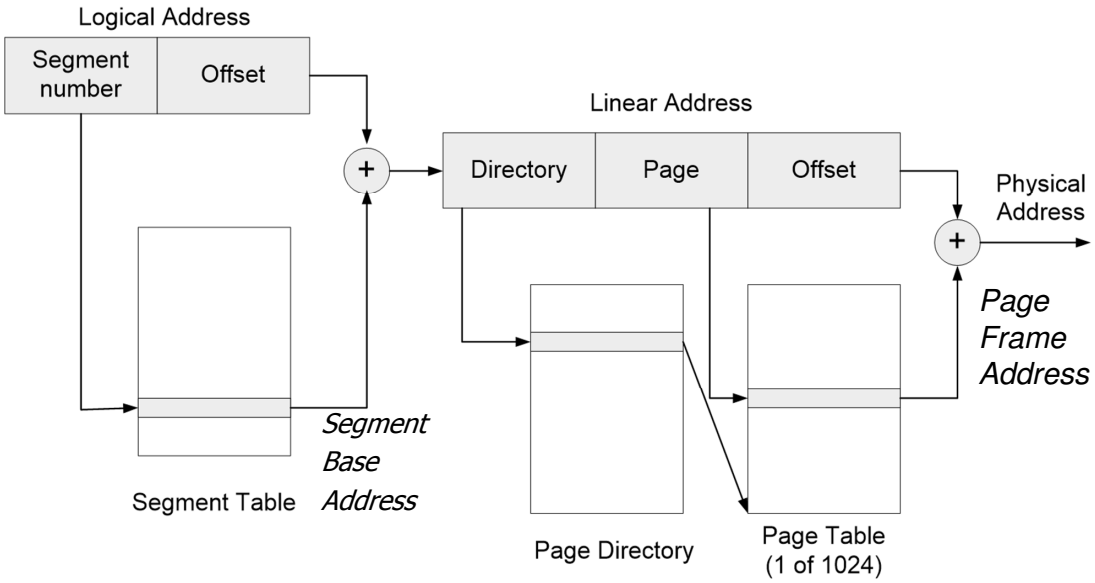


Self-study NOTES:
Paging





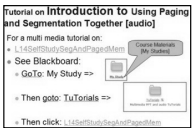
Self-study NOTES:
Segmentation and Paging



Self-study NOTES:

Problems in Address Translation

- The Page Directory Table and Page Table are held in memory.
- The Segment Table is held within processor.
- Thus, to access a memory location, 3 references to memory are required:
  1) One to the Page Directory;
  2) One to the Page Table; and [whether it is a]
  3) memory read or write to the physical memory (cache or main memory);
     - Potentially very slow.



Self-study NOTES:

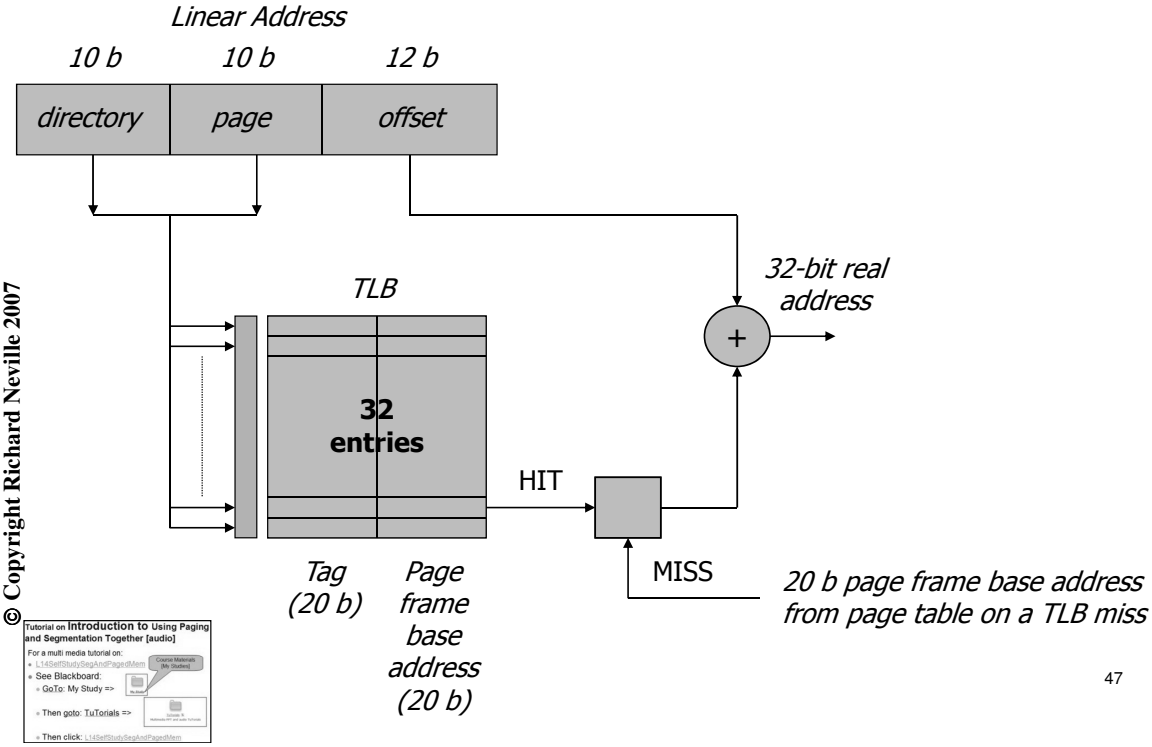
Translation Lookaside Buffer

- The Translation Lookaside Buffer (TLB) is a cache that contains the results of previously computed access to the tables;
  - So for a given linear address it holds the page table entry for that address.
- Thus, once a page is accessed, further linear addresses that refer to that page do not need the tables to be searched, the value is taken from the TLB.

Ref: URL http://www.isi.edu/mass/MuriloHomePage/Java/midterm402/midterm2.html
For a real time Applet sim. that is very visually informative; re. TLB's.

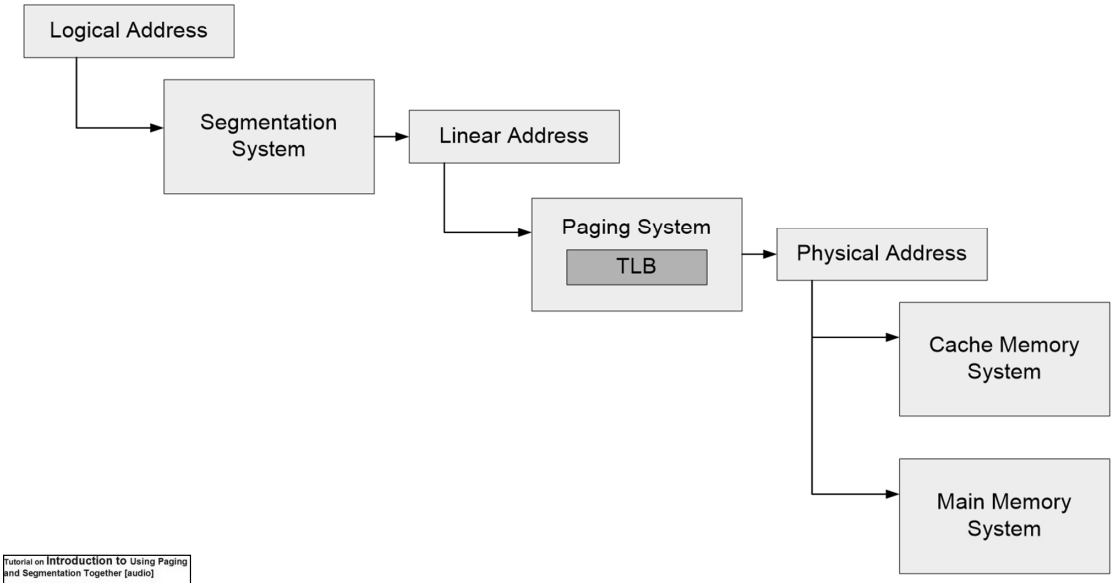
Self-study NOTES:

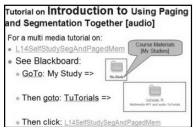
Translation Lookaside Buffer



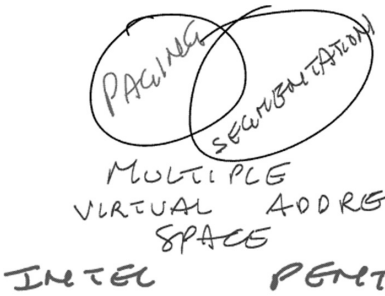
Self-study NOTES:

Virtual Memory with Cache Memory

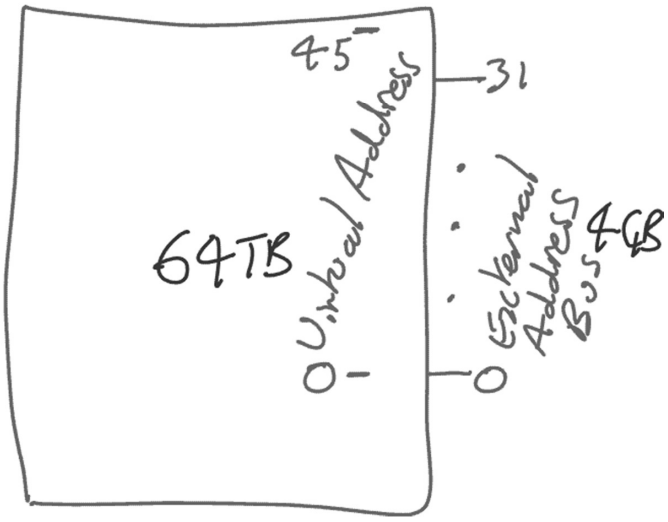




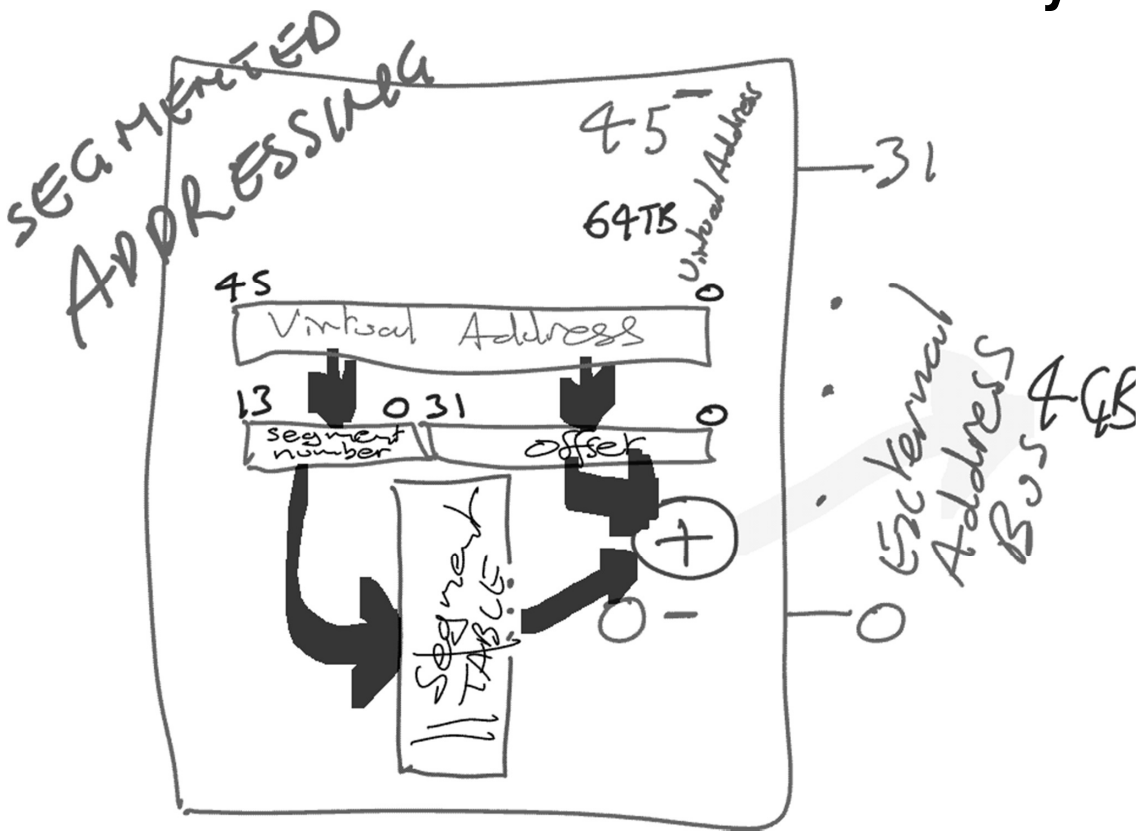
Summary



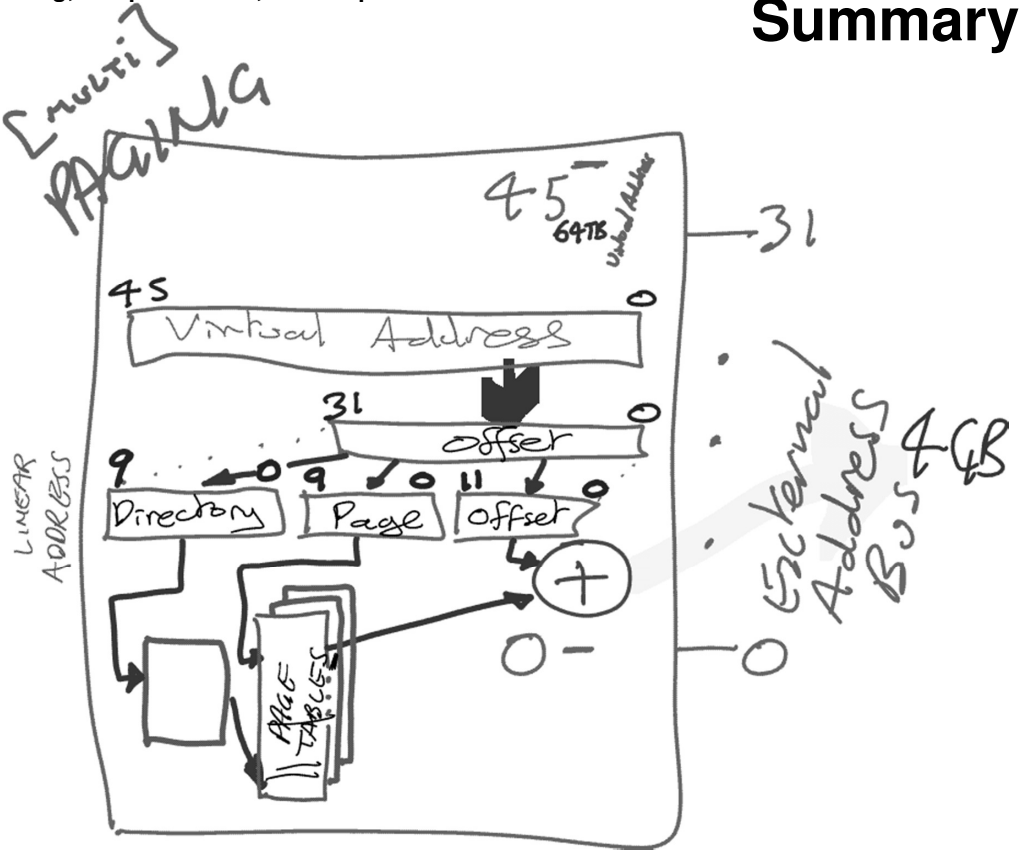
INTEL PENTIUM



Summary

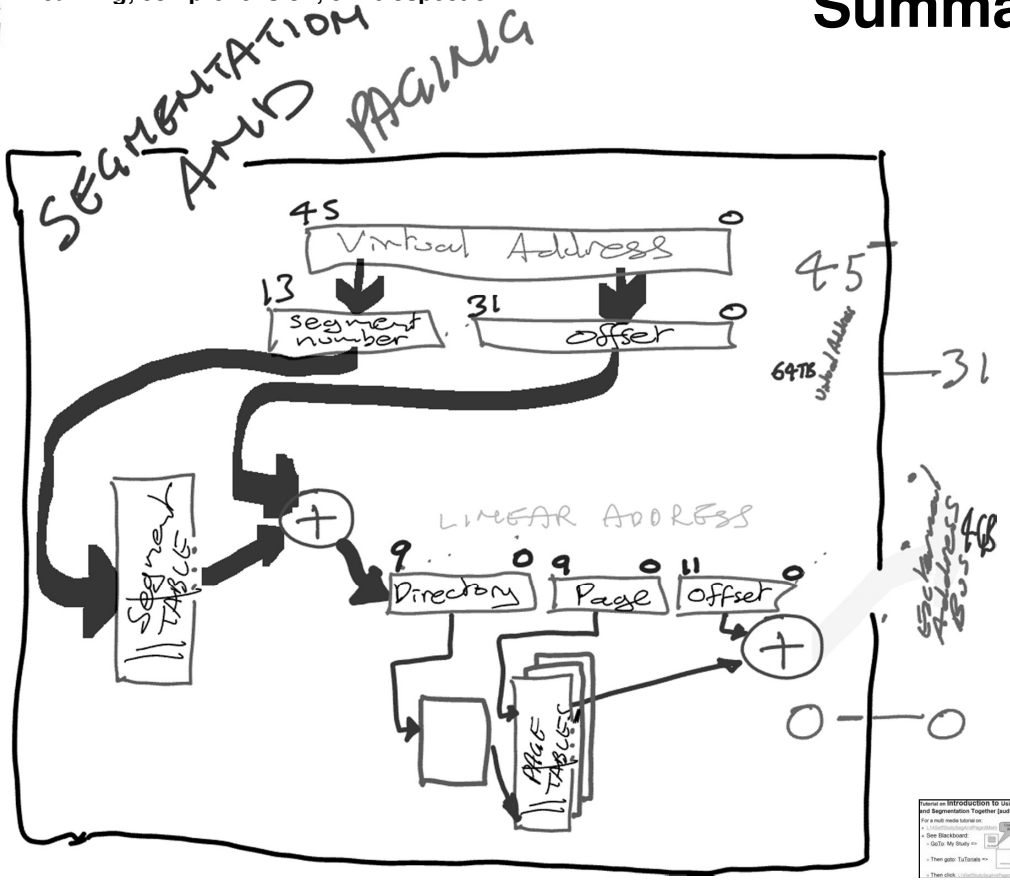


Summary

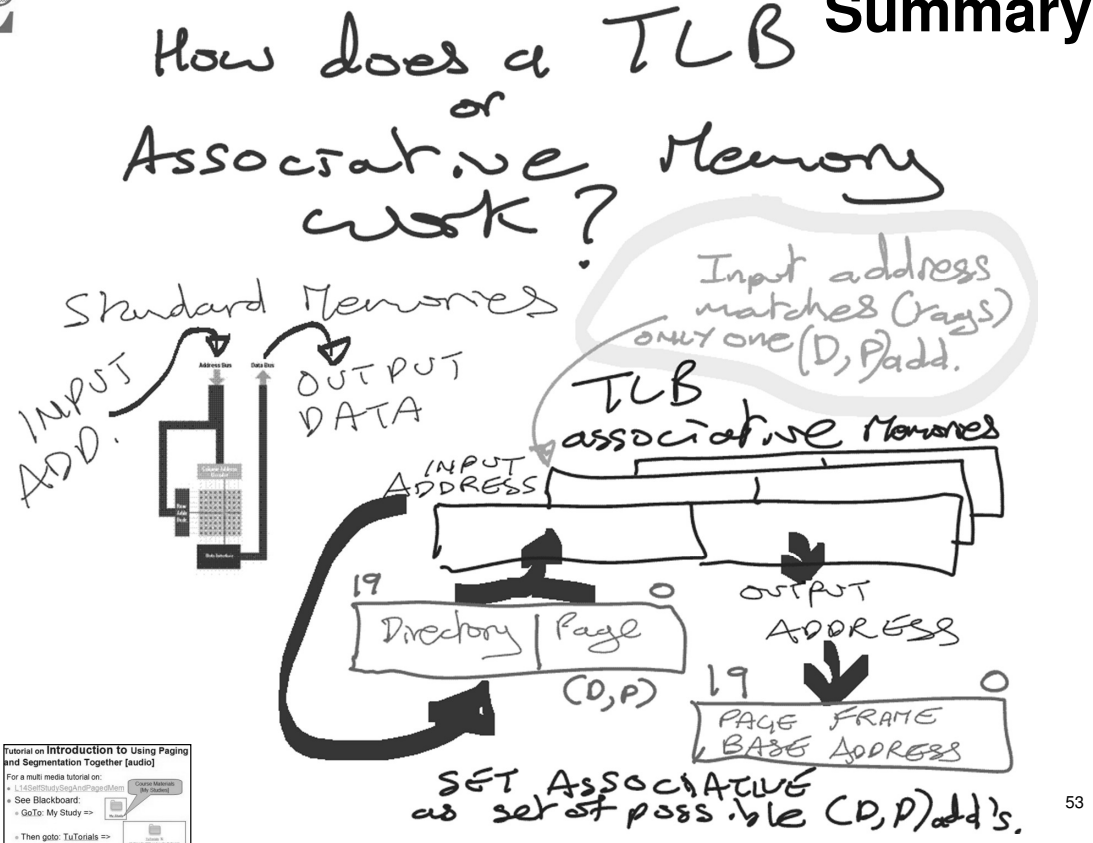


Learning; comprehension; & introspection

Summary

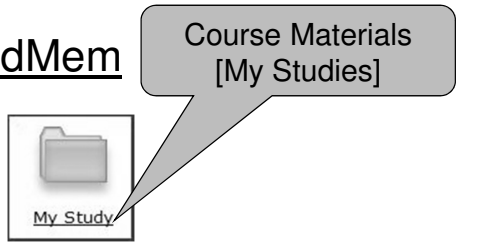


Summary



Tutorial on Introduction to Using Paging and Segmentation Together [audio]

- For a multi media tutorial on:
  - L14SelfStudySegAndPagedMem
  - See Blackboard:
    - GoTo: My Study =>
    - Then goto: TuTorials =>
    - Then click: L14SelfStudySegAndPagedMem



Glossary

- Why build a Glossary for each course unit you undertake?
  - It is imperative that the correct terminology [keywords] are utilised in context in your exam answers; this is so important that the lecturer has added glossaries to each paper copy of your lectures. It is of such importance that in your notes [prior to the start of the glossary] the following advice is given:
    - Each module you undertake uses its own jargon.
    - This can be a problem for new students, whom are trying to comprehend the new domain knowledge attached to a particular new module.
    - One way to get to know the new jargon is to build your own GLOSSARIES for each course module.
    - The glossary on the next few pages is a starting point for this module [unit].
    - Please feel free to add to the glossaries throughout the unit…
    - The glossary is full of potential exam questions of the form "define X" or "briefly explain X."
  - Please heed the advice in the future; even if your lecturers do not supply a glossary build your own as without knowledge of the appropriate terminology [keywords] when expounding your knowledge you will not be viewed as comprehending the details of any theory.

GLOSSARY

Using the on-line resources and any other resources compile a glossary of the terms below [PIPLINES: memory management]:

- Segmented →
- variably-sized →
- Access rights →
- Usage rights →
- Interfere →
- Segment Number →
- Offset →
- Segment Table →
- Segment resident →
- Segment used →
- Segment dirty →
- Segment size →

GLOSSARY

- segment physical base address →
- segment disk memory address →
- segment fault →
- External Fragmentation →
- Hole (in segmented memory) →
- Compact →
- shuffling →
- Best Fit →
- First fit →
- Programmer Visible →

GLOSSARY

- Block Replacement →
- Memory Use →
- Efficient Disk Traffic →
- multiple virtual address spaces →
- mapping →
- page directory →
- linear address →
- Translation →
- Lookaside →
- Buffer →
- Translation Lookaside Buffer →

GLOSSARY & AUX. DATA for expanded answers

- Dirty bit; how it is utilised:

  The dirty bit indicates when the memory has been modified. The dirty bit is a status bit which marks the block that has been modified. When a modification happens the bit is set. With the write-back strategy when the marked [dirty bit set] block is written back to secondary memory the dirty bit is reset; as a clean block replaces it.

Learning Resources 1

- [1] Structured Computer Organization, Andrew S Tannenbaum, Various Editions, Prentice Hall
  - Section 6.1
- **Descriptions [Theory] (in text books)**
- Remember the key issues, highlighted in GREEN, are the concepts to look for in any book:
  - Section on Intel 80x86: segment registers, segment addressing in chapter 5 the operating systems in: Chalk BS, Carter AT, Hind RW (2004) Computer Organisation and Architecture: An introduction 2nd Edition, Palgrave, ISBN 1-4039-0164-3 .
  - Section on pure translation lookaside buffers, segmentation, segmentation with paging & segmentation with paging the Intel Pentium in chapter 4 memory management in: Modern Operating Systems (MOS) 2nd Edition Andrew Tanenbaum.
  - Section on segmentation, Pentium: segmentation, translation lookaside buffer– in chapter 7 the Operating System Support in: Computer Organization and Architecture, Fifth Edition by William Stallings.
  - Section on segmentation, implementation of segmentation, virtual memory on the Pentium – in chapter 6 the Computer Systems Organization; & Parallel Computer Architectures in: Structured Computer Organization, 5/E, Andrew S. Tanenbaum, Vrije University, Amsterdam, The Netherlands, ISBN-10: 0131485210, ISBN-13: 9780131485211, Publisher: Prentice.

- **Web resources:**
  - Applet 'Virtual Memory Simulation;' available [on-line] @ http://www.isi.edu/mass/MuriloHomePage/Java/midterm402/midterm2.html
  - Virtual Memory Management; available [on-line] @ http://www.cs.mun.ca/~paul/cs3725/material/web/notes/node11.html
  - Virtual Memory available [on-line] @ http://courses.cs.vt.edu/csonline/OS/Lessons/VirtualMemory/index.html
  - 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

Questions

Introduction to Questions:

The set of questions are based on lecture 14.

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.

Questions

1. Question

Briefly explain the difference between a *page* and a *segment* in a virtual memory system.

Answer(s):

Questions

2. Question

Segmented and paged virtual memory systems are often combined. Explain how a 'segmented paged virtual memory' system is organised and explain the benefits of using such a system.

Answer(s):

Questions

3. Question

In the context of the Intel Pentium processor and paging the 32-bit linear address is broken into 3 fields. State the names of the three fields and give a description of each.

3. Answer

Answer(s):

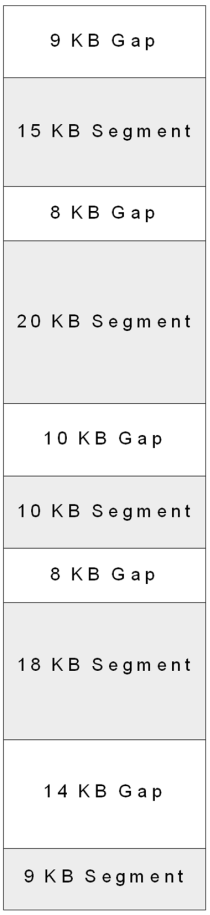
4. Question

- A computer system uses *segmented* virtual memory (no pages). The state of the memory at a given time is shown in the figure overleaf. Indicate what happens when a segment requiring 9KB of memory space is loaded using the following algorithms:
  1) Best Fit; and
  2) First Fit.

  State where the 9KB will be placed given 1) and 2).
- Note that it is assumed that the lowest address is at the bottom of the diagram.

Questions

- Present state of the memory for Question 4.



Lowest address is at the bottom of the diagram ⟶

Questions

4. Answer

1)

2)

Answer(s):

Revision Exercises

- Scan read Lecture 14's Questions.
  - Answer Lecture 14's Questions
    - Particularly those questions you had difficulties with when you first tried them.

Background Reading

[1] Computer Organisation and Architecture: An introduction, Chalk BS, Carter AT, Hind RW (2004) 2nd Edition, Palgrave, ISBN 1-4039-0164-3.

- Section on Intel 80x86: segment registers, segment addressing in chapter 5 the operating systems.

[2] Modern Operating Systems (MOS) 2nd Edition Andrew Tanenbaum.

- Section on pure translation lookaside buffers, segmentation, segmentation with paging & segmentation with paging the Intel Pentium in chapter 4 memory management.

[3] Computer Organization and Architecture, Fifth Edition by William Stallings

- Section on segmentation, Pentium: segmentation, translation lookaside buffer– in chapter 7 the Operating System Support.

[4] Structured Computer Organization, 5/E, Andrew S. Tanenbaum, Vrije University, Amsterdam, The Netherlands, ISBN-10: 0131485210, ISBN-13: 9780131485211, Publisher: Prentice.

- Section on segmentation, implementation of segmentation, virtual memory on the Pentium – in chapter 6 the Computer Systems Organization; & Parallel Computer Architectures in:.