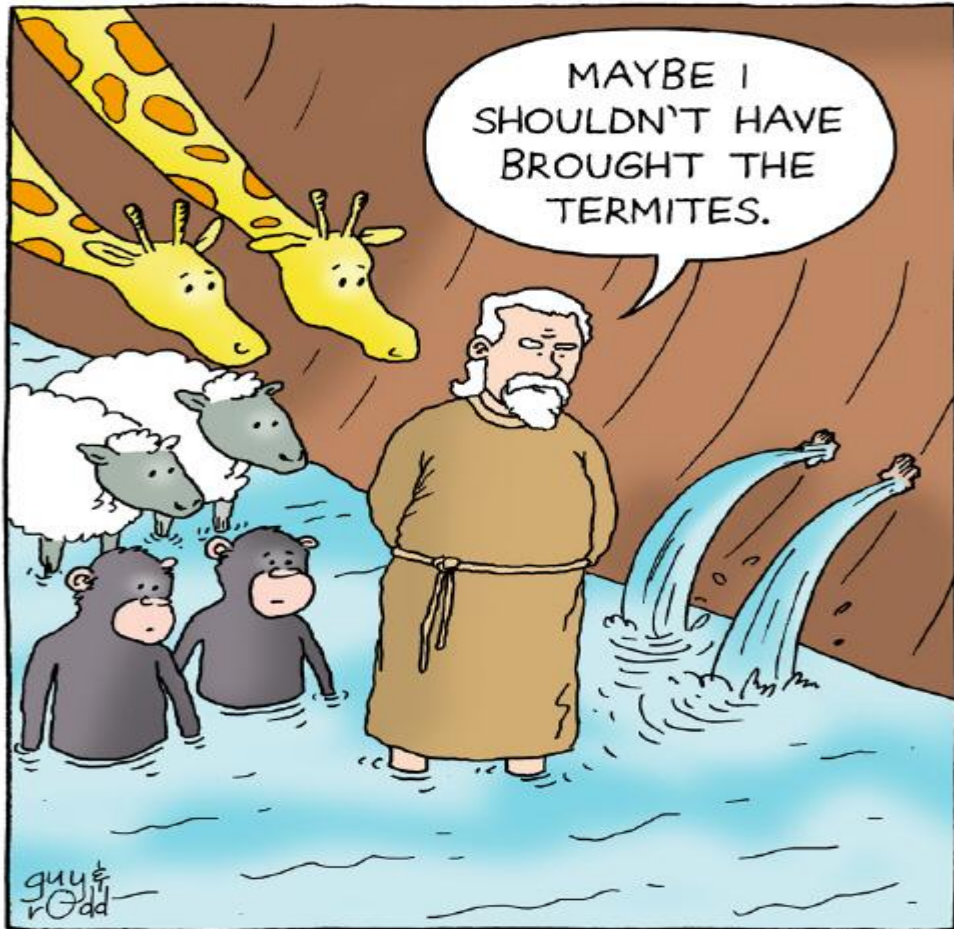


# Segmentation





MAYBE I  
SHOULDN'T HAVE  
BROUGHT THE  
TERMITES.

guy &  
rOod

# Overview

- History
- What is Paging?
- Paging Example
- Pros/Cons of Paging.
- What of Segmentation.
- Addressing Segments.
- Segmentation Example
- Pros/Cons of Segmentation.
- Video #1

# Overview cont.

- Segmentation vs. Paging
- Segmentation with Paging
- Multics - quick case study
- Implementation: Segmentation w/paging:  
MULTICS
- Video 2
- Summary
- Class Questions

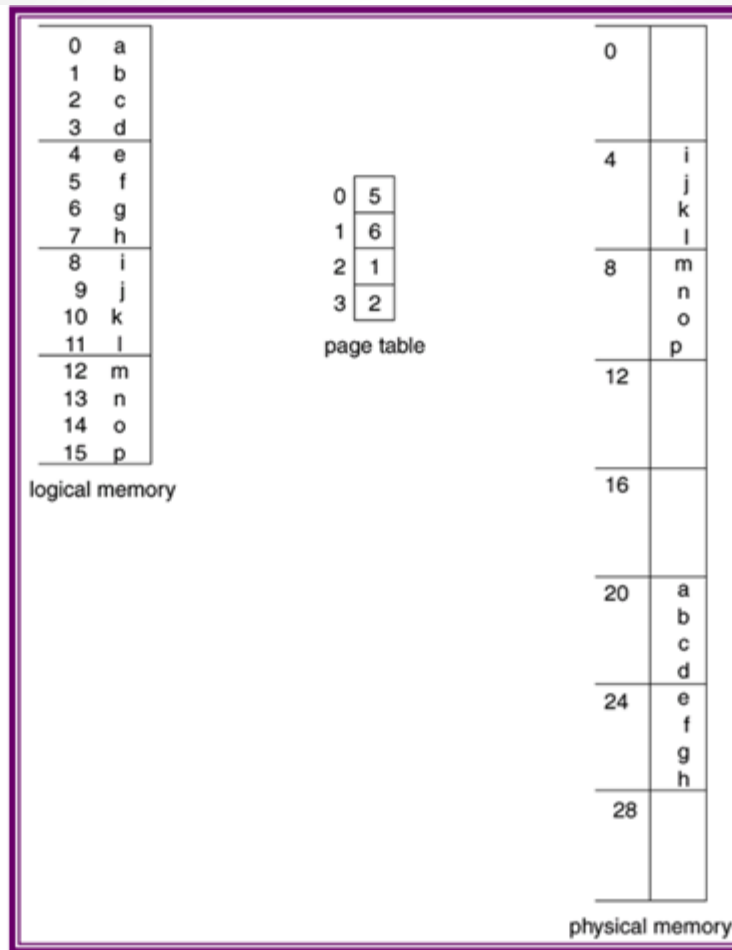
# History

- Virtual memory was developed in approximately 1959 – 1962, at the University of Manchester for the Atlas Computer, completed in 1962.
- In 1961, Burroughs released the B5000, the first commercial computer with virtual memory based on segmentation.

# What is Paging?

- During paging, the system divides the memory into pages .
- A page is a *physical* entity of a *fixed size*.
- Memory can be divided into large page sizes or small page sizes.
- A whole page must be used even if only holding a very small portion of data.

# Paging example



# Paging

## Small Pages:

- Large amounts of data cannot be held on a single page.
- Pages are frequently swapped in and out causing page faults.
- A page fault is what occurs when a page is too small to hold program
- Program thrashing
- Thrashing is when virtual memory is constant state of paging causing many exchanges of data in memory for data on disk.

## Large Pages:

- Require less swapping
- Memory is quickly depleted
- Under utilization of physical memory due to fragmentation



# Pros/Cons of Paging

## Advantages:

- Efficient memory usage
- Simple partition management due to discontinuous loading and fixed partition size
- No compaction necessary
- Easy to share pages

# Pros/Cons of Paging

## Disadvantages:

- Job Size  $\leq$  Memory Size
- Internal fragmentation (half the page size on the average)
- Need special hardware for address translation
- Some main memory space used for PMT's
- Address translation lengthens memory cycle times

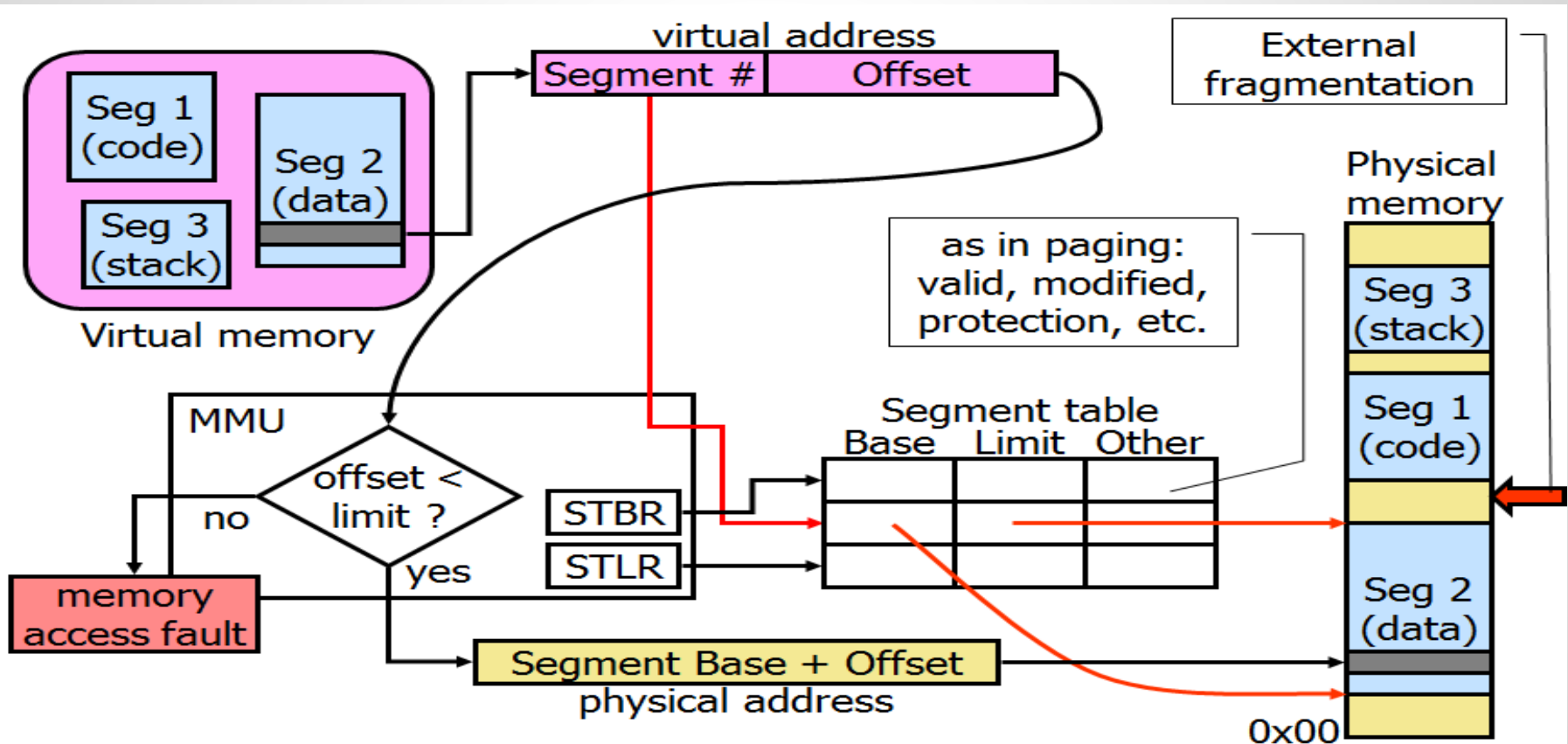
# Segmentation

A compromise was needed between large and small page sizes to maximize efficiency and improve performance.

# Segmentation

- Memory- management scheme that break the main memory in logical pieces called “***Segments***”.
- Segments are divisions of computer memory of ***variable size***.
- A program is a collection of segments. A segment is a logical unit such as:
  - main program, procedure, logical variables, global variables, symbols table, arrays...

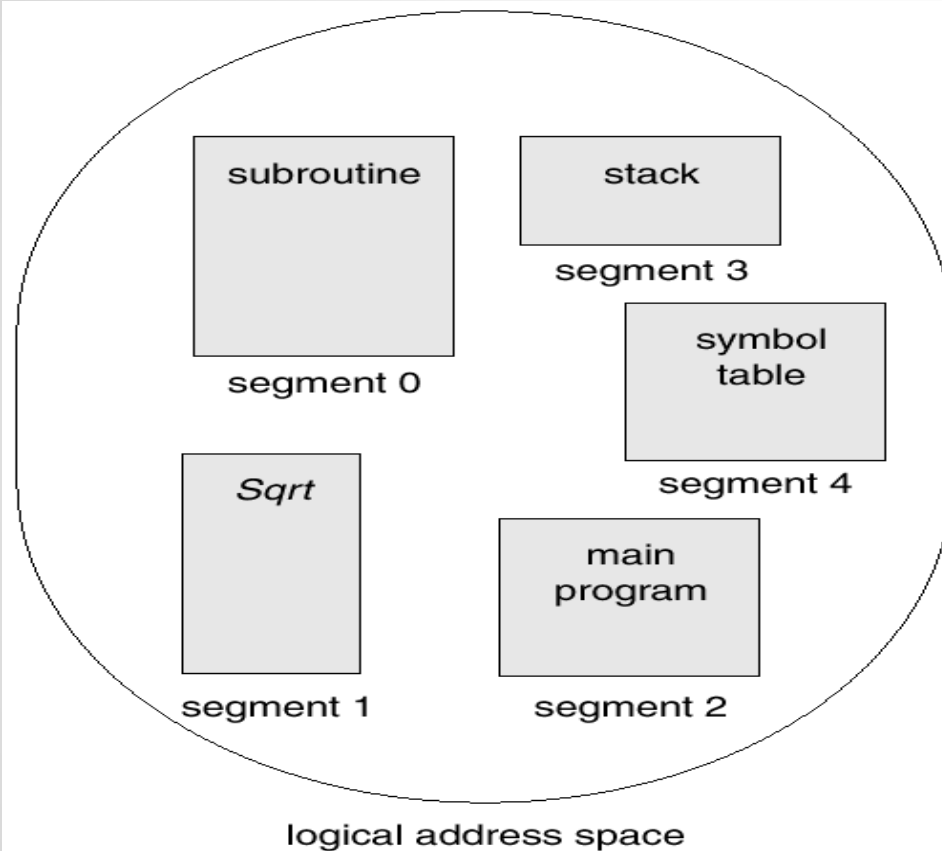
# What is Segmentation?



# Addressing Segments

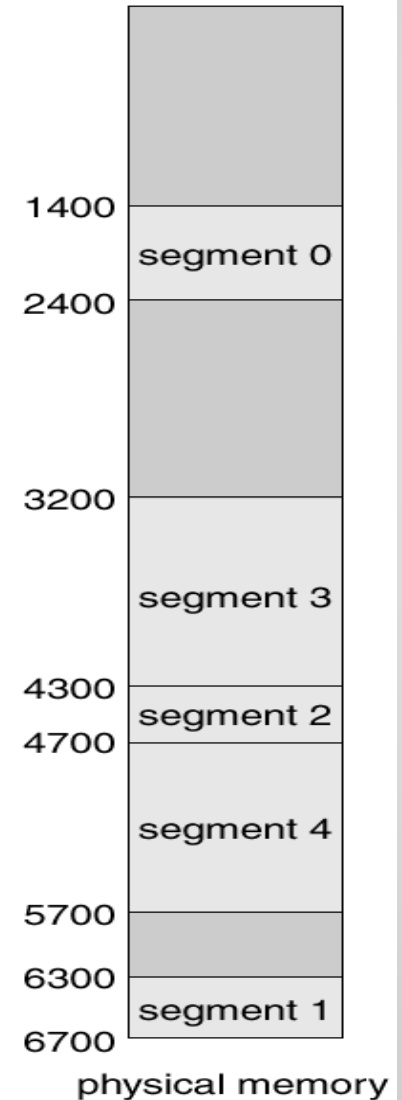
- Let's first assume no paging in the system
- User generates logical addresses
- These addresses consist of a segment number and an offset into the segment
- Use segment number to index into a table
- Table contains the physical address of the start of the segment
  - -often called the base address
- Add the offset to the base and generate the physical address
  - -before doing this, check the offset against the limit
  - -the limit is the size of the segment

# Example of Segmentation



	limit	base
0	1000	1400
1	400	6300
2	400	4300
3	1100	3200
4	1000	4700

segment table



# Pros/ Cons of Segmentation

## Advantages:

- No internal fragmentation **but** external fragmentation.
- May save memory if segments are very small and should not be combined into one page.
- Segment tables: only one entry per actual segment as opposed to one per page in VM
- Average segment size >> average page size

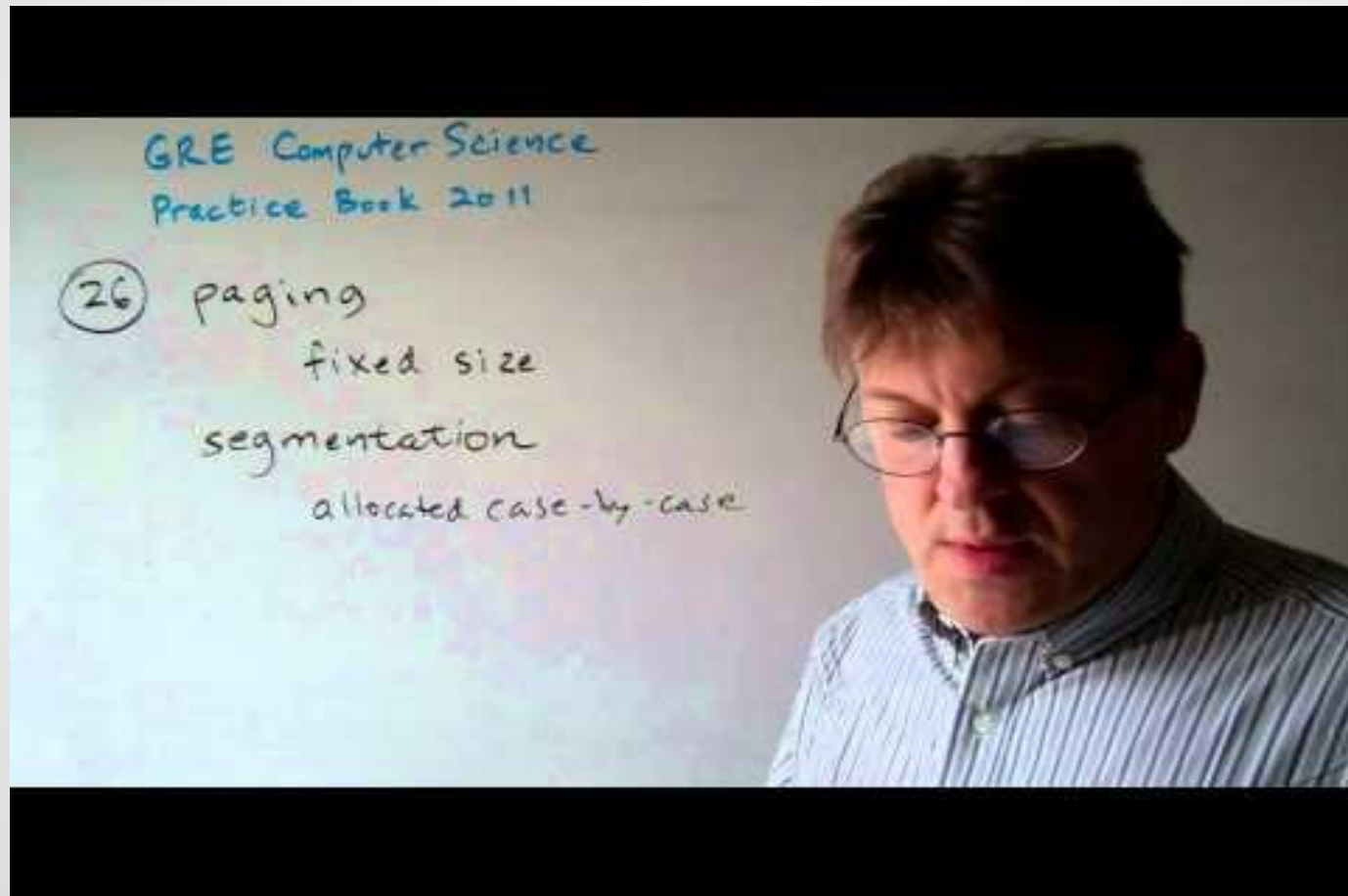


# Pros/ Cons of Segmentation

## Disadvantages:

- External fragmentation
- Costly memory management algorithms
  - Segmentation: find free memory area big enough (search!).
  - Paging: keep list of free pages, any page is ok (take first!).

# Video #1



# Paging vs. Segmentation

- Need the programmer be aware that this is being used?
- How many linear address spaces are there?
- Can the total address space exceed the size of physical memory?
- Is sharing of procedures between users facilitated?

# Segmentation with paging

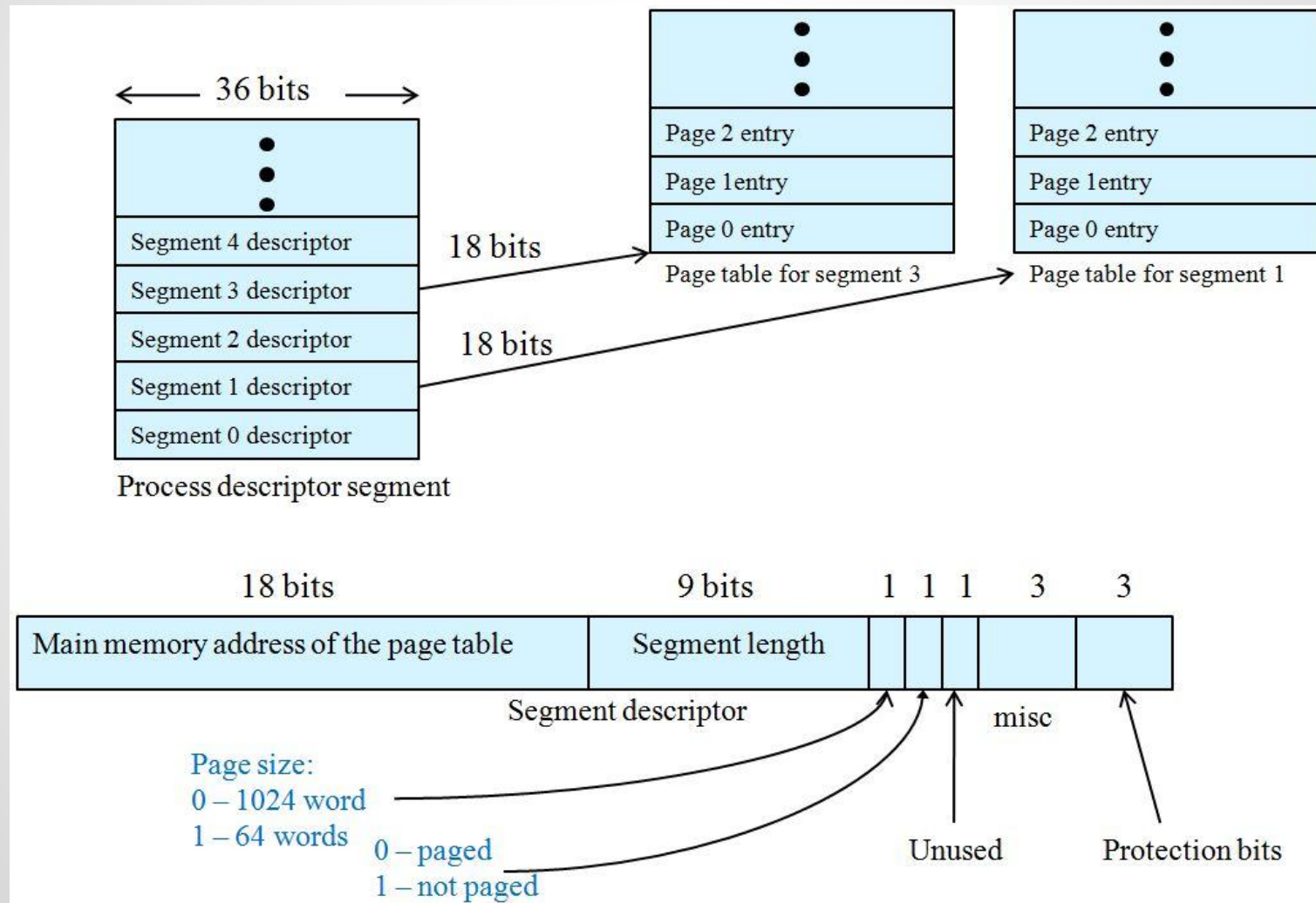
- Some modern processors allow usage of both, segmentation and paging alone or in a combination (Motorola 8030 and later, Intel 80386, 80486, Pentium) - the OS designers have a choice.



# Segmentation with paging: MULTICS cont.

- The MULTICS OS Ran on Honeywell computers
- Segmentation + paging
- Up to 218 segments
- Segment length up to 216 36-bit words
- Each program has a segments table (itself a segment)
- Each segment has a page table

# MULTICS data-structures

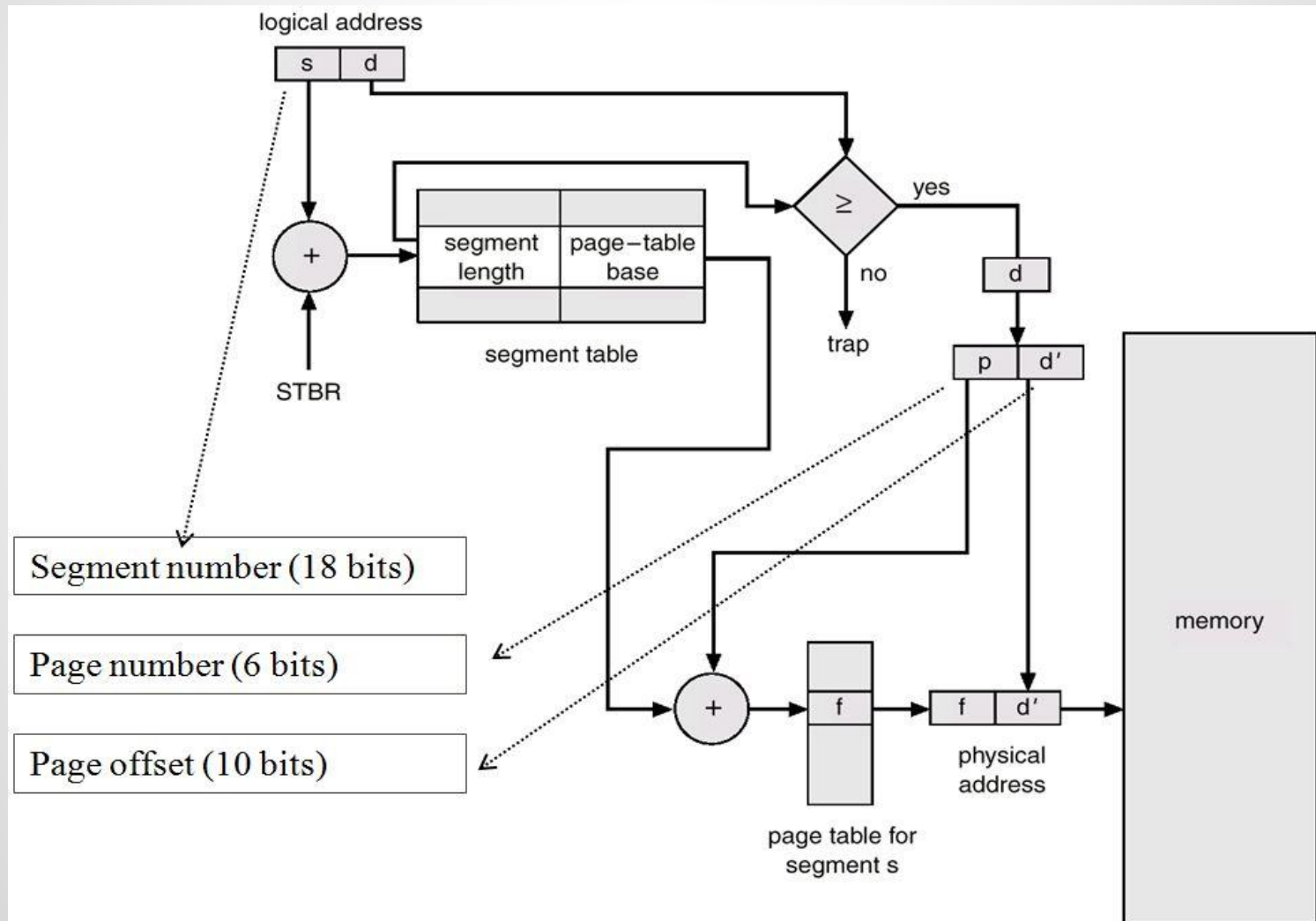


# MULTICS memory reference procedure

1. Use segment number to find segment descriptor  
*Descriptor segment is itself paged because it may be large. The descriptor-base-register points to its page table*
2. Check if segment's page table is in memory
  - if not a segment fault occurs
  - if there is a protection violation TRAP (fault)
3. page table entry examined, a page fault may occur
  - if page is in memory the start-of-page address is extracted from page table entry
4. offset is added to the page origin to construct main memory address
5. perform read/store etc.



# MULTICS Address Translation Scheme



# MULTICS TLB

Comparison field		Page frame	Protection	Age	Is this entry used?
Segment number	Virtual page				↓
4	1	7	Read/write	13	1
6	0	2	Read only	10	1
12	3	1	Read/write	2	1
					0
2	1	0	Execute only	7	1
2	2	12	Execute only	9	1

- Simplified version of the MULTICS TLB
- Existence of 2 page sizes makes actual TLB more complicated

# **MULTICS Additional checks during segment link (call)**

- Since segments are mapped to files, ACL (access control list) are checked with first access (open)
- Protection rings are called

**A very advanced architecture for 1970's !**

# Implementation

## Segmentation with paging: MULTICS

*Presenting the program .....*

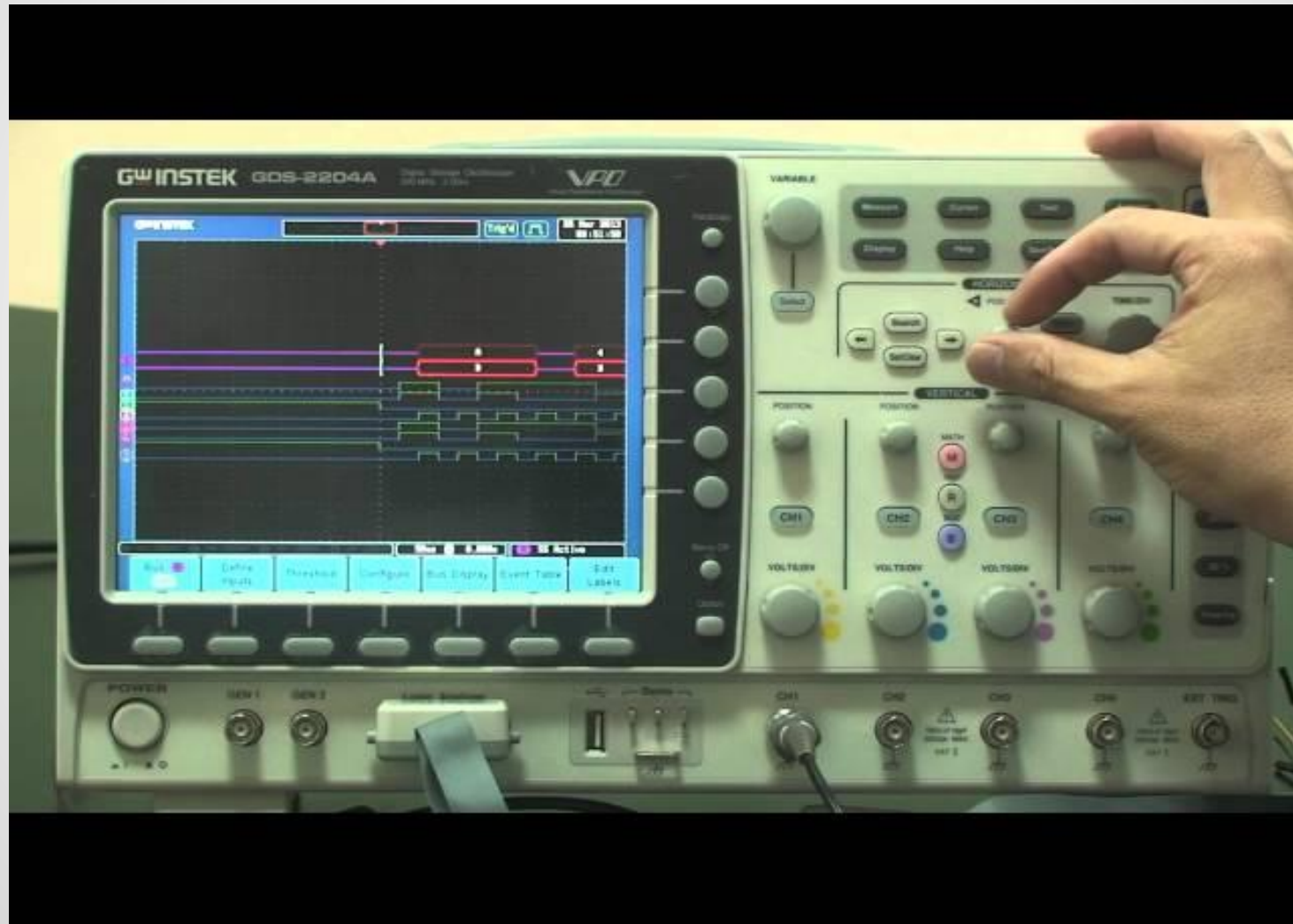
The image displays the MULTICS Memory Management interface, which is used for managing memory in a segmented and paged environment. The interface is divided into several sections:

- Control Panel:** Includes buttons for "Create Process" and "Open Page Files". It shows "Memory in MB" set to 100 and radio buttons for "Contiguous" (selected) and "Non-Contiguous".
- Process Table:** A table listing active processes and functions. The table has columns for ID, Type, Memory, Pages, and Timer.
- Performance Metrics:** A "Fast" to "Slow" slider and a "Memory Usage" gauge showing 0% to 100%.
- Memory Segments:** Three sections labeled "Code", "Data", and "Stack" with corresponding visual representations of memory segments.

In the foreground, a Notepad window titled "14 - Notepad" displays the following information:

```
File Edit Format View Help
ProcessId: 14
Page File Count: 1
Memory Usage: 28
Active Functions: 1
Process Lifetime: 16.4168671875
Process Color: java.awt.Color[r=125,g=197,b=253]
```

# Video 2



# Summary / Conclusion

- What is Paging?
- What is Segmentation?
- Examples of Paging and Segmentation.
- Pros/Cons of Paging and Segmentation.
- Two Videos.
- Segmentation vs. Paging.
- Segmentation with Paging.
- Multics.
- Implementation Segmentation with Paging:  
MULTICS.

**Class ...**

