Virtual Memory

CS3600

Spring 2022

Principles of virtual memory

- ***Virtual memory*** (***VM***) is a technique that allows the execution of processes that are not completely in memory.

- A paged VM creates a single large contiguous address space per process.

- A paged VM with segmentation creates multiple large address spaces per process, each of which is paged.

Demand Paging

- ***Demand paging*** is the principle of loading a page into memory only when the page is needed, rather than at the start of the execution.

- A ***present bit/valid bit*** is a binary flag in each page table entry that indicates whether the corresponding page is currently resident in memory.

- A ***page fault*** is an interrupt that occurs when a program attempts to reference a non-resident page. The interrupt triggers the operating system to find the page on disk, copy the page into a frame, and set the present bit to 1.

Q

- The process executes the code

```
   BR m         //Branch instruction
   ...
m: LDR R,x     //Load instruction ,'m' is the label ,'x' is a variable.
```

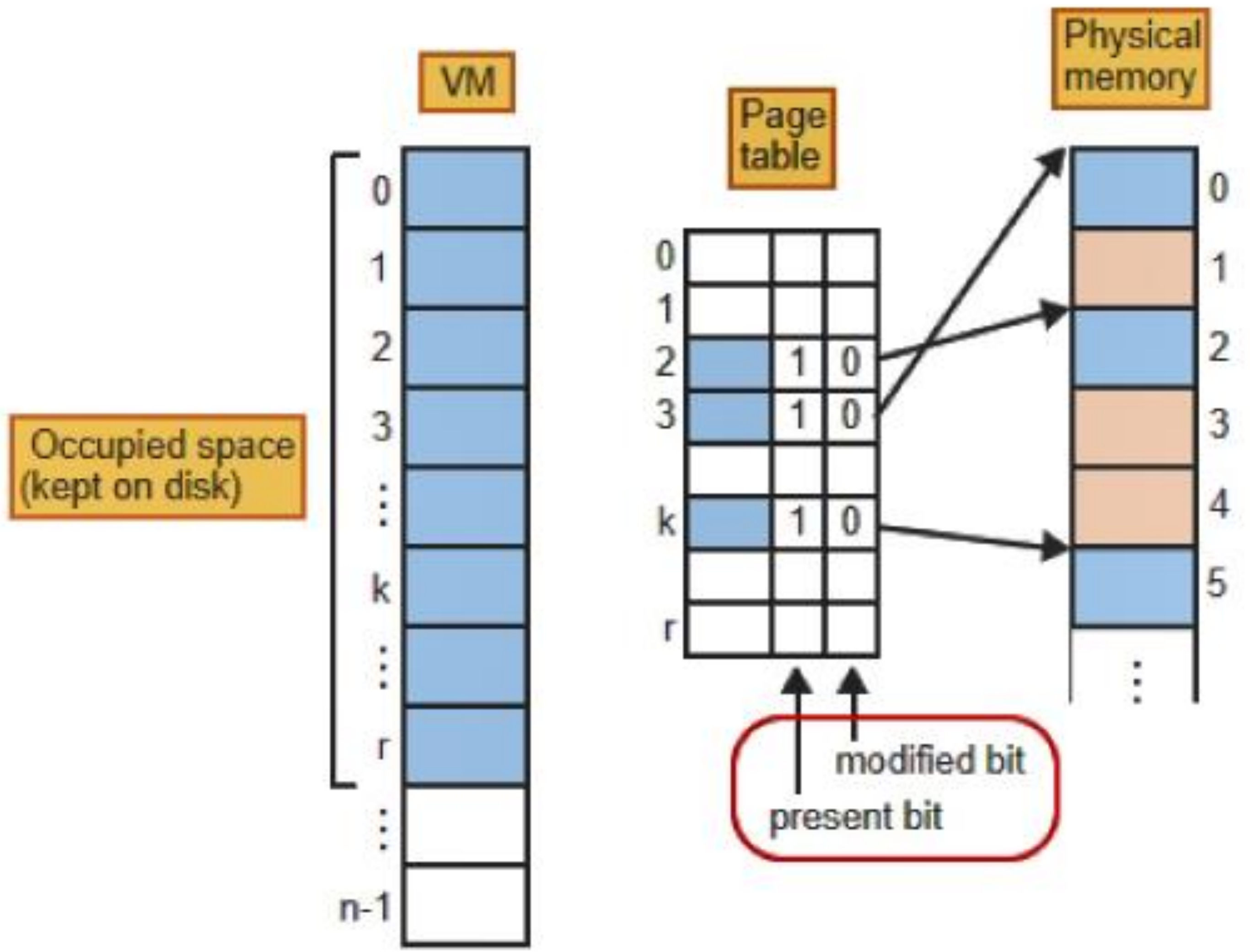
where the branch instruction is on a resident page and the instruction “LDR" loads the content of the variable x into a register R.

Will executing the code trigger page faults?

Page Replacement

- Virtual memory is typically much larger than the available physical memory. When all frames are occupied and a page fault occurs, one of the resident pages must be removed from memory to create a free frame.
- ***Page replacement*** is the act of overwriting a page in memory with a different page loaded from the disk when needed.
- A ***modifiy-bit*** (***m-bit or dirty bit***) is a binary flag in each page table entry that indicates whether the corresponding page has been modified during execution. The modified bit is set to 1 automatically by any instruction that stores data into the page and is used by the operating system to minimize the movement of data to disk.

Page Replacement

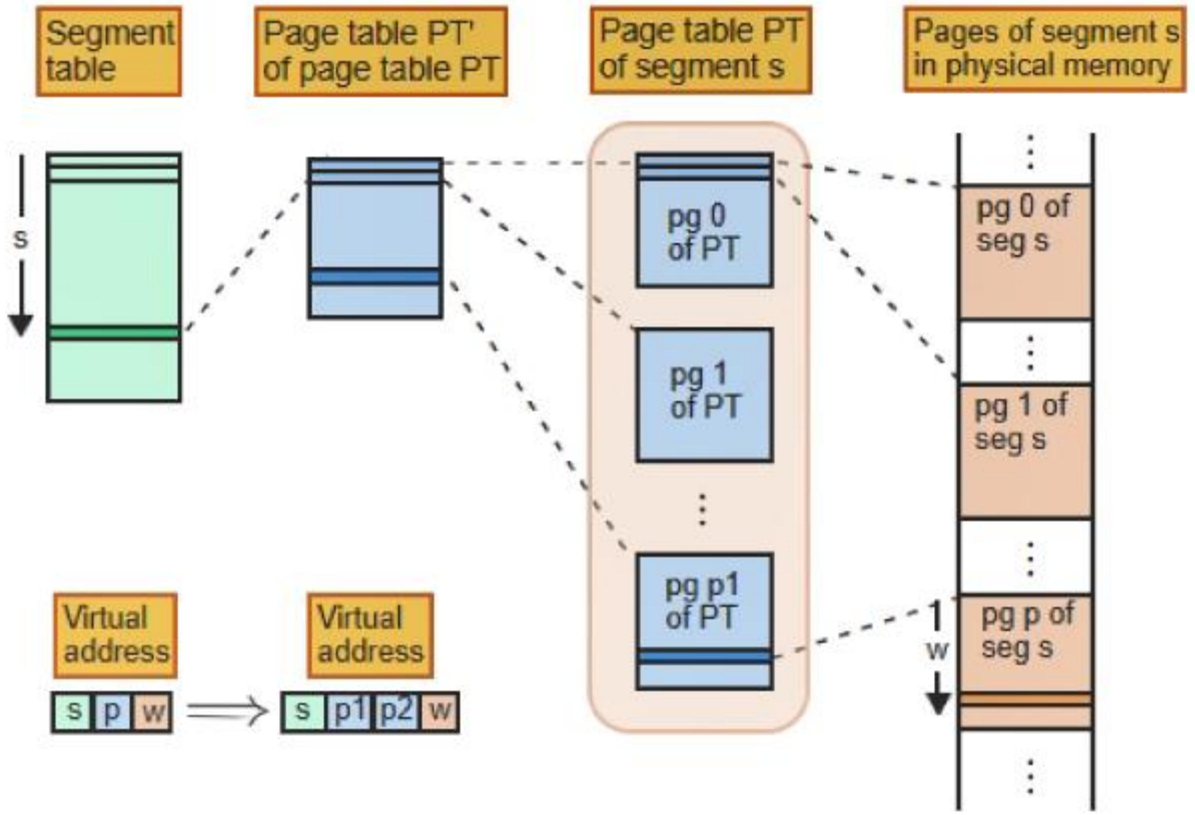


Paging of Tables

- Virtual memory can create address spaces much larger than the available physical memory.

- A segment table can be divided into pages by dividing the segment number s into two components, s1 and s2, and to use a new page table to keep track of the pages constituting the segment table. Similarly, a large page table can be divided into pages and kept track of by another higher-order page table.

Address translation with a paged page table.



Page replacement with fixed numbers of frames.

- Memory consists of a fixed number of frames. One approach to sharing the frames among multiple processes is to assign a fixed number of frames to each process. When a page fault occurs and no free frame is available, the algorithm must choose one of the resident pages for replacement.

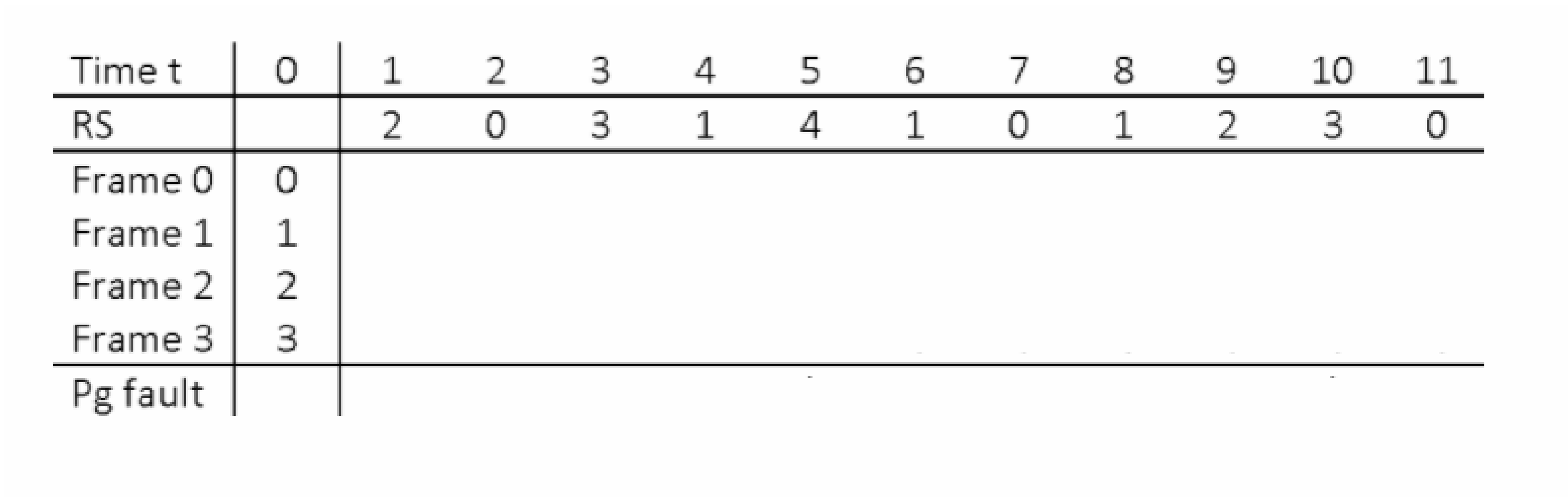
- A ***reference string*** is the sequence of page numbers referenced by an executing program during a given time interval. Reference strings are used to compare different page replacement algorithms by counting the number of page faults generated.

The optimal page replacement algorithm

- The ***optimal page replacement algorithm*** selects the page that will not be referenced for the longest time in the future.

- The algorithm is guaranteed to generate the smallest number of page faults for any reference string. However, the algorithm is unrealizable because the replacement decisions depend on future references, which are unknown at runtime. But the algorithm provides a useful lower bound on the number of page faults for comparison with other algorithms.

The optimal replacement algorithm



Time t	0	1	2	3	4	5	6	7	8	9	10	11
RS		2	0	3	1	4	1	0	1	2	3	0
Frame 0	0											
Frame 1	1											
Frame 2	2											
Frame 3	3											
Pg fault												

Q

A memory with 2 page frames contains pages 5 and 3. For an optimal replacement algorithm

The first page fault will occur at time _____.

Page _____ will be replaced by the first page fault.

Time	1	2	3	4	5	6	7
RS	**5**	**5**	**5**	**1**	**3**	**5**	**3**

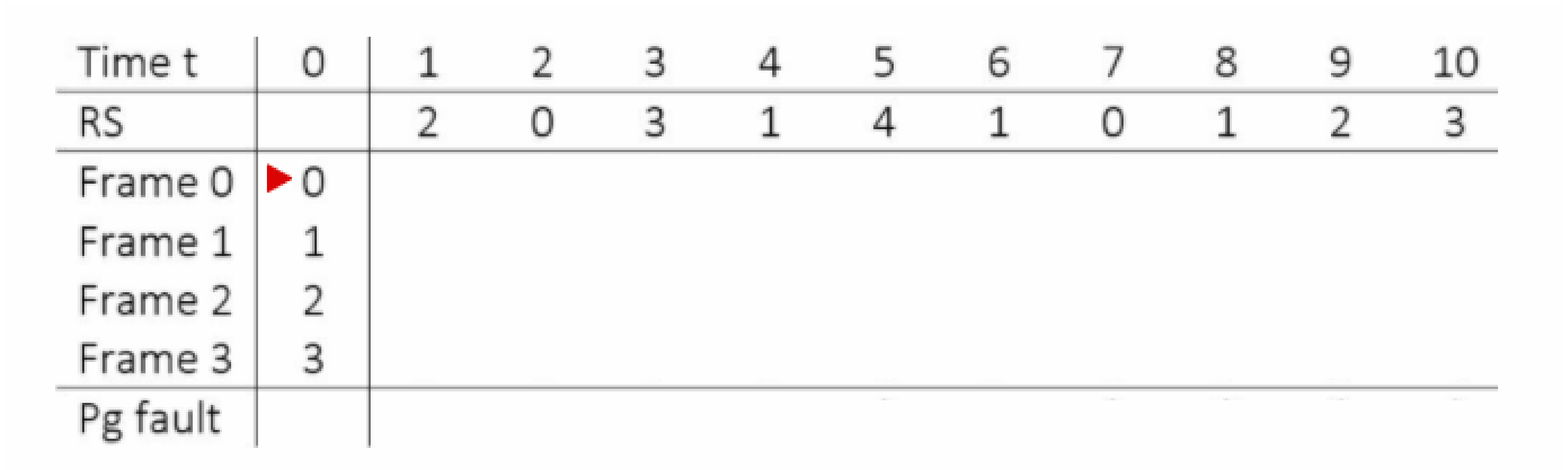
The FIFO page replacement algorithm

- The ***FIFO page replacement algorithm*** selects the page that has been resident in memory for the longest time.

- The algorithm is easy to implement, requiring only a single pointer to designate the oldest page in memory. When the page is replaced, the pointer is advanced to the next frame modulo n.

- The FIFO algorithm takes advantage of the principle of locality. Except for branch instructions, which constitute only a small percentage of all instructions, execution is sequential. As execution advances sequentially through the code, the likelihood of referencing a page used in the distant past diminishes with time. Similarly, many large data structures are processed in sequential order.

FIFO replacement algorithm



Time t	0	1	2	3	4	5	6	7	8	9	10
RS		2	0	3	1	4	1	0	1	2	3
Frame 0	▶ 0										
Frame 1	1										
Frame 2	2										
Frame 3	3										
Pg fault											

Q

- Memory has 3 frames containing pages 0, 1, and 2, where 0 is the oldest page. RS = {0 3 0 1}
- When page 3 is referenced, FIFO will replace page _____.
- The total number of page faults for the RS is _____

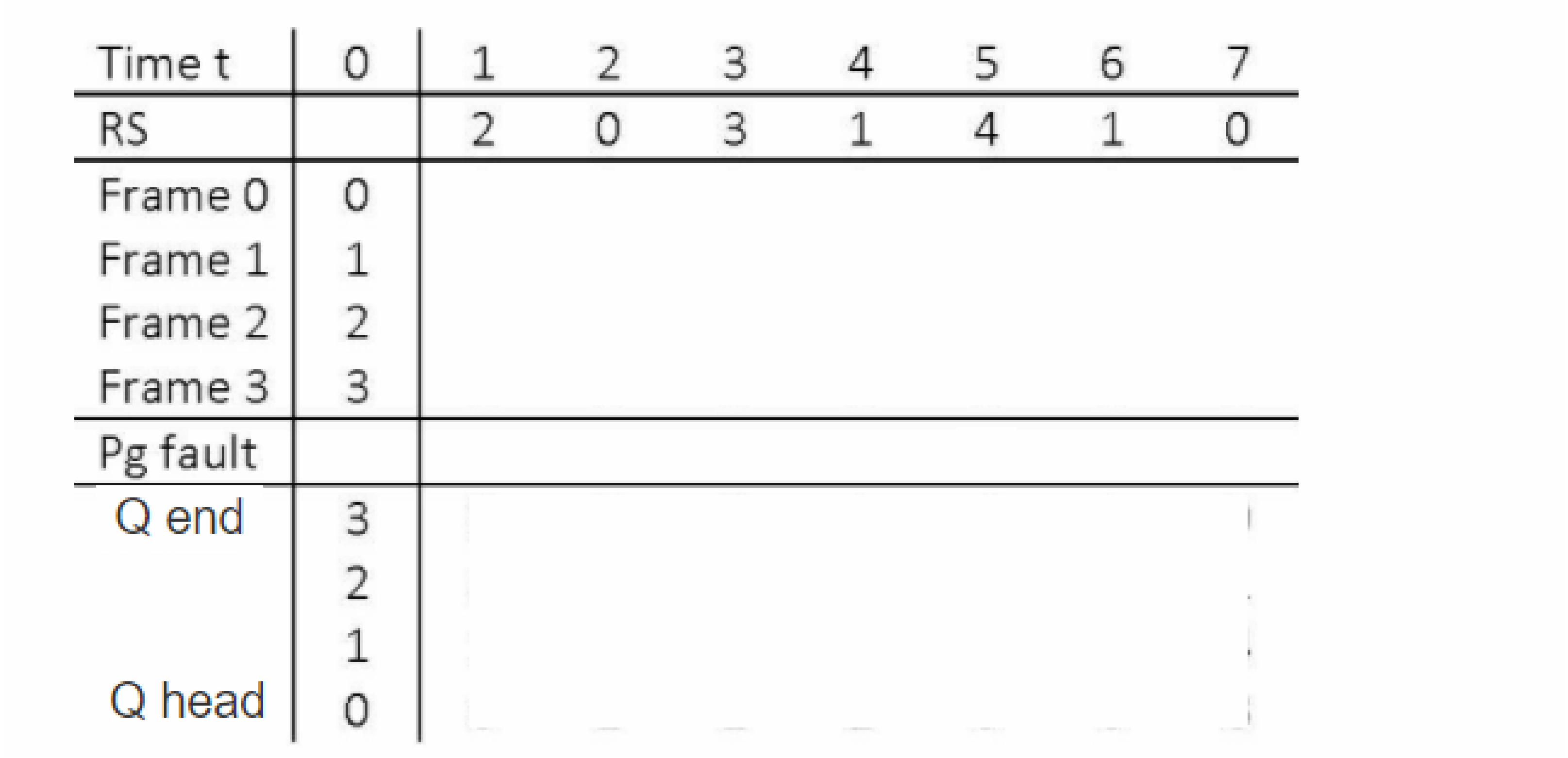
The least-recently-used page replacement algorithm (LRU)

- FIFO exploits the principle of locality to some degree but fails to recognize that a program frequently returns to pages referenced in the distant past. FIFO replaces the oldest resident pages even if the pages have been referenced again recently and thus are likely to be referenced again in the near future.

- The ***least-recently-used page replacement algorithm*** (***LRU***) selects the page that has not been referenced for the longest time.

- The implementation requires a queue of length n, where n is the number of memory frames. The queue contains the numbers of all resident pages.

LRU Algorithm



Time t	0	1	2	3	4	5	6	7
RS		2	0	3	1	4	1	0
Frame 0	0							
Frame 1	1							
Frame 2	2							
Frame 3	3							
Pg fault								
Q end	3							
	2							
	1							
Q head	0							

Q

- With 4 frames and RS = {1 2 3 4 2 }, the queue for LRU replacement will at the end contain the sequence of pages _____ .

- Worksheet 10