Operating Systems Concepts

Virtual Memory: Performance and Analysis

CS 4375, Fall *2025*
Instructor: MD Armanuzzaman (***Arman***)
marmanuzzaman@utep.edu
October 27, 2025

Summary

- Virtual Memory:
  - Demand Paging
  - Page Faults
  - Page Replacement
  - Page Replacement Algorithms
    - FIFO
    - Optimal Algorithm
    - LRU & LRU Approximations
    - Counting Algorithms

Agenda

- Virtual Memory
  - Page Replacement Algorithms
    - FIFO
    - Optimal Algorithm
    - LRU & LRU Approximations
    - Counting Algorithms
  - Thrashing
  - Performance of Demand Paging

Page Replacement Algorithms - Exercise

Consider the following page reference string:

1, 2, 3, 4, 4, 3, 2, 1, 5, 6, 2, 1, 2, 3, 7, 8, 3, 2, 1, 5

Assuming 4 memory frames and **LRU-8 bit** page replacement algorithm:

- How many page faults, page hits, and page replacements would occur?
- Show your page assignments to frames

Referenced Page		1	2	3	4	4	3	2	1	5	6	2	1	2	3	7	8	3	2	1	5
Frame-1																					
Frame-2																					
Frame-3																					
Frame-4																					

# of page faults: **# of page hits:** **# of page replacements:**

Page Replacement Algorithms - Exercise

		Initial	1	2	3	4	4	3	2	1	5	6
Memory frames + LRU bits	#1	-	1	1	1	1	1	1	1	1	1	1
	bits	0000 0000	1000 0000	0100 0000	0010 0000	0001 0000	0000 1000	0000 0100	0000 0010	1000 0001	0100 0000	0010 0000
	#2	-	-	2	2	2	2	2	2	2	2	2
	bits	0000 0000	0000 0000	1000 0000	0100 0000	0010 0000	0001 0000	0000 1000	1000 0100	0100 0010	0010 0001	0001 0000
	#3	-	-	-	3	3	3	3	3	3	3	6
	bits	0000 0000	0000 0000	0000 0000	1000 0000	0100 0000	0010 0000	1001 0000	0100 1000	0010 0100	0001 0010	1000 0000
	#4	-	-	-	-	4	4	4	4	4	5	5
	bits	0000 0000	0000 0000	0000 0000	0000 0000	1000 0000	1100 0000	0110 0000	0011 0000	0001 1000	1000 0000	0100 0000

		Cont.	2	1	2	3	7	8	3	2	1	5
Memory frames + LRU bits	#1	1	1	1	1	1	1	8	8	8	8	5
	bits	0010 0000	0001 0000	1000 1000	0100 0100	0010 0010	0001 0001	1000 0000	0100 0000	0010 0000	0001 0000	1000 0000
	#2	2	2	2	2	2	2	2	2	2	2	2
	bits	0001 0000	1000 1000	0100 0100	1010 0010	0101 0001	0010 1000	0001 0100	0000 1010	1000 0101	0100 0010	0010 0001
	#3	6	6	6	6	6	7	7	7	7	1	1
	bits	1000 0000	0100 0000	0010 0000	0001 0000	0000 1000	1000 0000	0100 0000	0010 0000	0001 0000	1000 0000	0100 0000
	#4	5	5	5	5	3	3	3	3	3	3	3
	bits	0100 0000	0010 0000	0001 0000	0000 1000	1000 0000	0100 0000	0010 0000	1001 0000	0100 1000	0010 0100	0001 0010

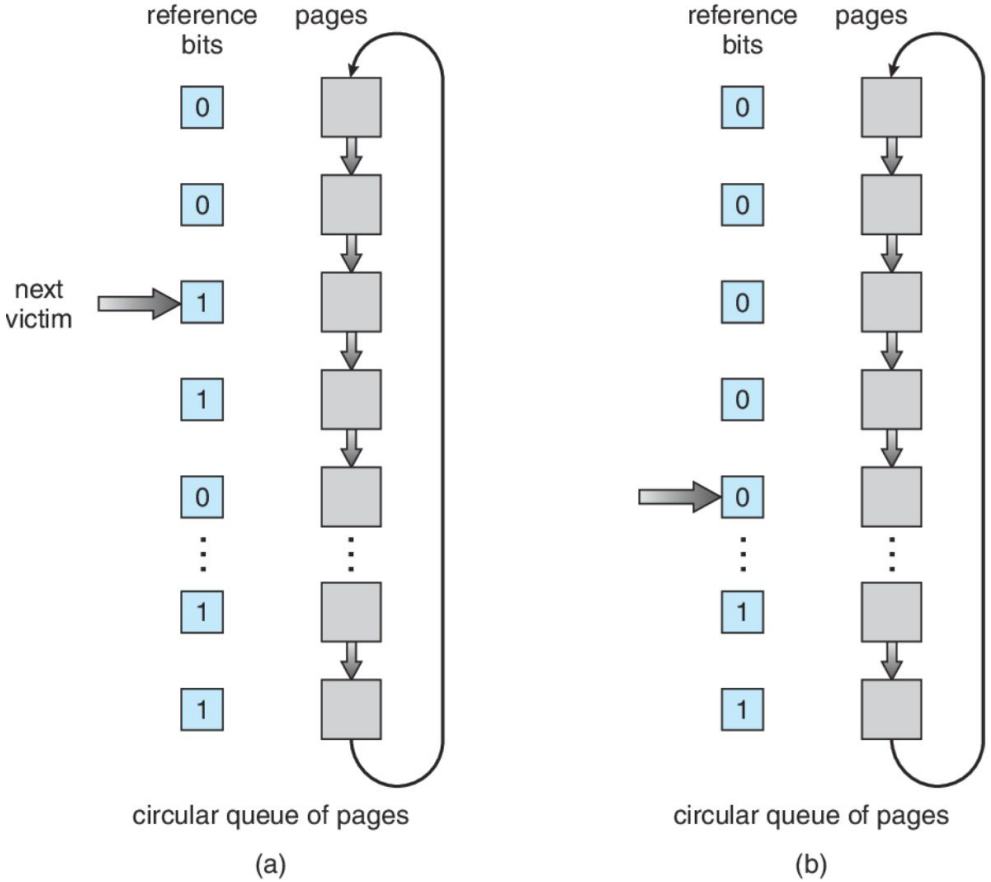
Page Replacement Algorithms - Counting Algorithms

- Keep a counter of the number of references that have been made to each page
- Least Frequently Used (LFU)
  - Replaces page with smallest count
- Most Frequently Used (MFU)
  - Based on the argument that the page with the smallest count was probably just brought in and has yet to be used

Page Replacement Algorithms - Second Chance

- LRU-Clock algorithm, also known as Not Recently Used (NRU) or Second Chance
  - Replace page that is old enough
  - Logically, arrange all physical page frames in a big circle (clock)
    - A circular linked list
  - A clock hand is used to select a good LRU candidate
    - Sweep through the pages in circular order like a clock
    - If reference bit is off, it hasn't been used recently → we have a victim!
    - If the reference bit is on, turn it off and go to next one → second chance
  - Arm moves quickly when pages are needed
  - Low overhead if we have plenty of memory
  - We can add more clock hands to improve

Page Replacement Algorithms - Second Chance



Page Replacement Algorithms - Exercise

Consider the following page reference string:

1, 2, 3, 4, 4, 3, 2, 1, 5, 6, 2, 1, 2, 3, 7, 8, 3, 2, 1, 5

Assuming 4 memory frames and LRU-Clock page replacement algorithm:

1. When a page is brought to the memory, reference bit is initialized to 0
2. Advance the victim pointer only if you need to find a victim to replace.

Referenced Page		1	2	3	4	4	3	2	1	5	6	2	1	2	3	7	8	3	2	1	5
Frame-1																					
Frame-2																					
Frame-3																					
Frame-4																					

# of page faults: **# of page hits:** **# of page replacements:**

Performance of Demand Paging

- **p:** Page fault rate $0 \leq p \leq 1.0$

  If **p = 0**, no page faults If **p = 1**, every reference is a fault

- M: memory access time
- T_{miss} : $T_{fault\_overhead} + [\ T_{swap\_out}\ ] + T_{swap\_in} + T_{restart}$
- T_{swap_out} : Depends on **dirty/not-dirty** page ratio → Find weighted average
- Effective Access Time (EAT):

$$EAT = (Hit\ Rate \times Hit\ Time) + (Miss\ Rate \times Miss\ Time)$$

$$EAT = (1 - p)\ M + p\ (T_{fault\_overhead} + [\ T_{swap\_out}\ ] + T_{swap\_in} + T_{restart} + M)$$

$$\text{Ignore overheads} \rightarrow EAT = (1 - p)\ M + p\ ([\ T_{swap\_out}\ ] + T_{swap\_in} + M)$$

$$EAT = M + p\ ([\ T_{swap\_out}\ ] + T_{swap\_in})$$

$$EAT = M + p\ (T_{effective\_swap\_out} + T_{swap\_in})$$

Performance of Demand Paging - Example

- M: memory access time = 1 us (1 ns $\ll$ 1 us $\ll$ 1 ms $\ll$ 1s)
- 50% of the time the page that is being replaced has been modified and therefore needs to be swapped out
- Swap Page Time = 10 ms = 10,000 us (Assume for either read or write)
- What is EAT (based on page fault ratio)?

Performance of Demand Paging - Example

- M: memory access time = 1 us (1 ns ≪ 1 us ≪ 1 ms ≪ 1s)
- 50% of the time the page that is being replaced has been modified and therefore needs to be swapped out
- Swap Page Time = 10 ms = 10,000 us (Assume for either read or write)
- What is EAT (based on page fault ratio)?

$$EAT = (Hit\ Rate \times Hit\ Time) + (Miss\ Rate \times Miss\ Time)$$

$$EAT = (1 - p)\ M + p\ ([\ T_{swap\_out}\ ] + T_{swap\_in} + M)$$

$$EAT = M + p\ (T_{effective\_swap\_out} + T_{swap\_in})$$

$$EAT = M + p\ (½ \times 10{,}000 + 10{,}000)$$

$$EAT = 1 + 15{,}000 \times p$$

Performance of Demand Paging - Example

- M: memory access time = 1 us (1 ns ≪ 1 us ≪ 1 ms ≪ 1s)
- 50% of the time the page that is being replaced has been modified and therefore needs to be swapped out
- Swap Page Time = 10 ms = 10,000 us (Assume for either read or write)
- What is EAT (based on page fault ratio)?

$$EAT = 1 + 15{,}000 \times p$$

- What if 1 out of 1000 memory accesses cause a page fault?

Performance of Demand Paging - Example

- M: memory access time = 1 us (1 ns ≪ 1 us ≪ 1 ms ≪ 1s)
- 50% of the time the page that is being replaced has been modified and therefore needs to be swapped out
- Swap Page Time = 10 ms = 10,000 us (Assume for either read or write)
- What is EAT (based on page fault ratio)?

$$EAT = 1 + 15{,}000 \times p$$

- What if 1 out of 1000 memory accesses cause a page fault?

$$EAT = 1 + 15{,}000 \times p \rightarrow EAT = 16 \text{ us} \rightarrow 1500\% \text{ degradation !!}$$

- What if we only want 30% performance degradation?

Performance of Demand Paging - Example

- M: memory access time = 1 us (1 ns ≪ 1 us ≪ 1 ms ≪ 1s)
- 50% of the time the page that is being replaced has been modified and therefore needs to be swapped out
- Swap Page Time = 10 ms = 10,000 us (Assume for either read or write)
- What is EAT (based on page fault ratio)?

$$EAT = 1 + 15{,}000 \times p$$

- What if 1 out of 1000 memory accesses cause a page fault?

$$EAT = 1 + 15{,}000 \times p \rightarrow EAT = 16 \text{ us} \rightarrow 1500\% \text{ degradation !!}$$

- What if we only want 30% performance degradation?

Exercise 1

Consider a paging system with the page table stored in memory.

a) If a memory reference takes 200 ns, how long does a paged memory reference take?

b) If we add associative registers, and 75 percent of all page-table references are found in the associative registers, what is the effective memory reference time? (Assume that finding a page-table entry in the associative registers takes zero time, if the entry is there.)

Exercise 1

Consider a paging system with the page table stored in memory.

a) If a memory reference takes 200 ns, how long does a paged memory reference take?

400 ns; 200 ns to access the page table and 200 ns to access the word in memory.

b) If we add associative registers, and 75 percent of all page-table references are found in the associative registers, what is the effective memory reference time? (Assume that finding a page-table entry in the associative registers takes zero time, if the entry is there.)

EAT = (0.75 x 200) + (0.25 x 400) = **250 ns**

Exercise 2-a

Consider a demand paging system where 40% of the page table is **stored in the registers,** and the rest needs to be addressed from the memory. Assume any reference to the memory takes 100 ns, and 20% of the time the page that is being replaced has its **dirty bit** set to 1. Swapping a page in takes 1000 us and swapping out a page takes 2000 us.

(1 ns $\ll$ 1 us $\ll$ 1 ms $\ll$ 1s)

a) How long does a paged memory reference take?

Exercise 2-a

Consider a demand paging system where 40% of the page table is **stored in the registers,** and the rest needs to be addressed from the memory. Assume any reference to the memory takes 100 ns, and 20% of the time the page that is being replaced has its **dirty bit** set to 1. Swapping a page in takes 1000 us and swapping out a page takes 2000 us.

(1 ns ≪ 1 us ≪ 1 ms ≪ 1s)

a) How long does a paged memory reference take?

Hit + Miss
(0.4 x M) + (0.6 x 2M) = 1.6 M = 160 ns = 0.160 us
or
Extra + Always
(0.6 x M) + M = 60 + 100 = 160 ns = 0.160 us

Exercise 2-b

Consider a demand paging system where 40% of the page table is **stored in the registers,** and the rest needs to be addressed from the memory. Assume any reference to the memory takes 100 ns, and 20% of the time the page that is being replaced has its **dirty bit** set to 1. Swapping a page in takes 1000 us and swapping out a page takes 2000 us.

(1 ns $\ll$ 1 us $\ll$ 1 ms $\ll$ 1s)

b) What is the Effective Access Time (EAT) if the page fault ratio is 0.001?

Exercise 2-b

Consider a demand paging system where 40% of the page table is **stored in the registers,** and the rest needs to be addressed from the memory. Assume any reference to the memory takes 100 ns, and 20% of the time the page that is being replaced has its **dirty bit** set to 1. Swapping a page in takes 1000 us and swapping out a page takes 2000 us.

(1 ns ≪ 1 us ≪ 1 ms ≪ 1s)

b) What is the Effective Access Time (EAT) if the page fault ratio is 0.001?

EAT = (1-p) x (Average Hit) + p x (Average Miss)
= (1-p) x 0.160 + p x ((0.80) x 1000 + (0.2) x 3000 + 0.160)
or = (1-p) x 0.160 + p x ((0.2) x 2000 + 1000 + 0.160)
= 0.160 + p x 1400
= 0.160 + 1/1000 x 1400 = 1.56 us

Exercise 2-b

Consider a demand paging system where 40% of the page table is **stored in the registers,** and the rest needs to be addressed from the memory. Assume any reference to the memory takes 100 ns, and 20% of the time the page that is being replaced has its **dirty bit** set to 1. Swapping a page in takes 1000 us and swapping out a page takes 2000 us.

(1 ns ≪ 1 us ≪ 1 ms ≪ 1s)

b) What is the Effective Access Time (EAT) if the page fault ratio is 0.001?

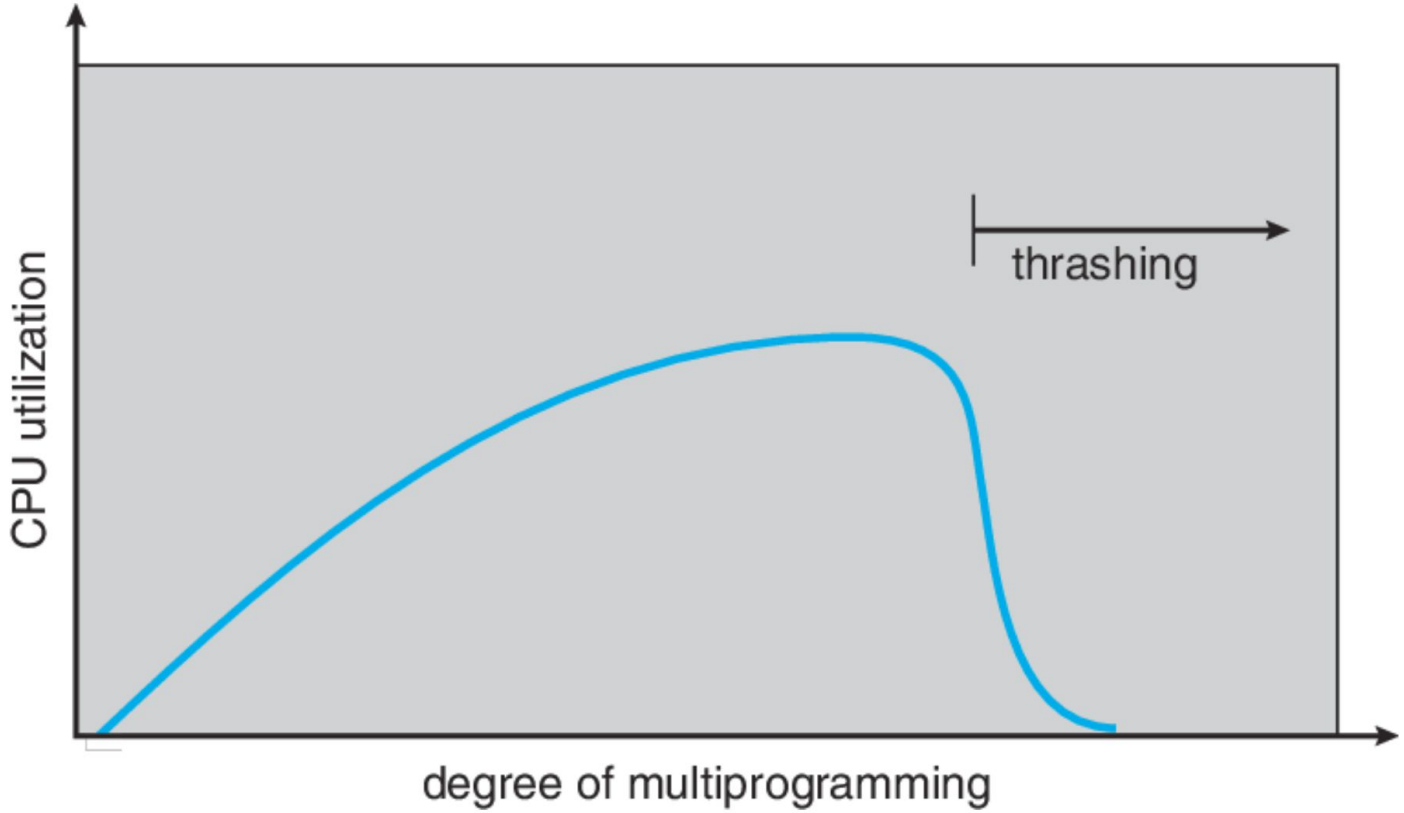
EAT = M + p ([Average Swapping overhead])
= M + p ([Average Swap out] + Swap in)
= 0.160 + p x ((0.2)x2000 + 1000)
= 0.160 + p x 1400
= 0.160 + 1/1000 x 1400 = 1.56 us

Thrashing

- If a process does not have “enough” frames, the page-fault rate is very high. This leads to replacement of active pages which will be needed soon again.
- Thrashing happens when a process is busy swapping pages in and out
- This will in turn, cause low CPU utilization.
- Operating system may think that it needs to increase the degree of multiprogramming
- Another process added to the system
- …

- Thrashing occurs when a computer's virtual memory resources are overused.

Thrashing



Exercise 3-a

Consider a demand-paged computer system where the degree of multiprogramming is currently fixed at four. The system was recently measured to determine utilization of CPU and the paging disk. The results are one of the following alternatives. For each case, what is happening (in one phrase)? Can you increase the degree of multiprogramming to increase the CPU utilization?

a) CPU utilization 96 percent; disk utilization 4 percent.

Exercise 3-a

Consider a demand-paged computer system where the degree of multiprogramming is currently fixed at four. The system was recently measured to determine utilization of CPU and the paging disk. The results are one of the following alternatives. For each case, what is happening (in one phrase)? Can you increase the degree of multiprogramming to increase the CPU utilization?

a) CPU utilization 96 percent; disk utilization 4 percent.

CPU utilization is sufficiently high to leave things alone (there are already sufficient processes running to keep the CPU busy); increasing the degree of multiprogramming may decrease the CPU utilization.

Exercise 3-b

Consider a demand-paged computer system where the degree of multiprogramming is currently fixed at four. The system was recently measured to determine utilization of CPU and the paging disk. The results are one of the following alternatives. For each case, what is happening (in one phrase)? Can you increase the degree of multiprogramming to increase the CPU utilization?

b) CPU utilization 10 percent; disk utilization 95 percent.

Exercise 3-b

Consider a demand-paged computer system where the degree of multiprogramming is currently fixed at four. The system was recently measured to determine utilization of CPU and the paging disk. The results are one of the following alternatives. For each case, what is happening (in one phrase)? Can you increase the degree of multiprogramming to increase the CPU utilization?

b) CPU utilization 10 percent; disk utilization 95 percent.

Thrashing is occurring. We cannot increase the CPU utilization

Exercise 3-c

Consider a demand-paged computer system where the degree of multiprogramming is currently fixed at four. The system was recently measured to determine utilization of CPU and the paging disk. The results are one of the following alternatives. For each case, what is happening (in one phrase)? Can you increase the degree of multiprogramming to increase the CPU utilization?

c) CPU utilization 12 percent; disk utilization 2 percent.

Exercise 3-c

Consider a demand-paged computer system where the degree of multiprogramming is currently fixed at four. The system was recently measured to determine utilization of CPU and the paging disk. The results are one of the following alternatives. For each case, what is happening (in one phrase)? Can you increase the degree of multiprogramming to increase the CPU utilization?

c) CPU utilization 12 percent; disk utilization 2 percent.

Both CPU and disk utilization are low, and CPU is obviously underutilized. We should increase the degree of multiprogramming to increase CPU utilization.

Exercise 4-a

Consider a demand-paging system with the following time-measured utilization:

CPU utilization 18%

Paging disk 96%

Other I/O devices 6%

For each of the following, say whether it will (or is likely to) improve CPU utilization. Answer with YES or NO or LIKELY, and justify your answers.

a) Install a faster CPU

Exercise 4-a

Consider a demand-paging system with the following time-measured utilization:

CPU utilization 18%

Paging disk 96%

Other I/O devices 6%

For each of the following, say whether it will (or is likely to) improve CPU utilization. Answer with YES or NO or LIKELY, and justify your answers.

a) Install a faster CPU

NO, a faster CPU reduces the CPU utilization further since the CPU will spend more time waiting for a process to enter in the ready queue.

Exercise 4-b

Consider a demand-paging system with the following time-measured utilization:

CPU utilization 18%

Paging disk 96%

Other I/O devices 6%

For each of the following, say whether it will (or is likely to) improve CPU utilization. Answer with YES or NO or LIKELY, and justify your answers.

b) Install a bigger paging disk

Exercise 4-b

Consider a demand-paging system with the following time-measured utilization:

CPU utilization 18%

Paging disk 96%

Other I/O devices 6%

For each of the following, say whether it will (or is likely to) improve CPU utilization. Answer with YES or NO or LIKELY, and justify your answers.

b) Install a bigger paging disk

NO, the size of the paging disk does not affect the amount of memory that is needed to reduce the page faults.

Exercise 4-c

Consider a demand-paging system with the following time-measured utilization:

CPU utilization 18%

Paging disk 96%

Other I/O devices 6%

For each of the following, say whether it will (or is likely to) improve CPU utilization. Answer with YES or NO or LIKELY, and justify your answers.

c) Decrease the degree of multiprogramming

Exercise 4-c

Consider a demand-paging system with the following time-measured utilization:

CPU utilization 18%

Paging disk 96%

Other I/O devices 6%

For each of the following, say whether it will (or is likely to) improve CPU utilization. Answer with YES or NO or LIKELY, and justify your answers.

c) Decrease the degree of multiprogramming

YES, by suspending some of the processes, the other processes will have more frames in order to bring their pages in them, hence reducing the page faults.

Exercise 4-d

Consider a demand-paging system with the following time-measured utilization:

CPU utilization 18%

Paging disk 96%

Other I/O devices 6%

For each of the following, say whether it will (or is likely to) improve CPU utilization. Answer with YES or NO or LIKELY, and justify your answers.

d) Install more main memory

Exercise 4-d

Consider a demand-paging system with the following time-measured utilization:

CPU utilization 18%

Paging disk 96%

Other I/O devices 6%

For each of the following, say whether it will (or is likely to) improve CPU utilization. Answer with YES or NO or LIKELY, and justify your answers.

d) Install more main memory

Likely, more pages can remain resident and do not require paging to or from the disks (i.e. would depend on the page replacement algorithm you are using and the page reference sequence).

Acknowledgements

- “Operating Systems Concepts” book and supplementary material by A. Silberschatz, P. Galvin and G. Gagne
- “Operating Systems: Internals and Design Principles” book and supplementary material by W. Stallings
- “Modern Operating Systems” book and supplementary material by A. Tanenbaum
- R. Doursat and M. Yuksel from University of Nevada, Reno
- Farshad Ghanei from Illinois Tech
- T. Kosar and K. Dantu from University at Buffalo

Announcement

- Homework 3
  - Due on Today, 27th at 11.59PM
- Next Class
  - Discussion about homework 4
  - Release homework 4