FIT2100 Tutorial #6 Memory Management, I/O Management, and Disk Scheduling (Suggested Solutions) Week 11 Semester 2 2017

Dr Jojo Wong
Lecturer, Faculty of IT.
Email: Jojo.Wong@monash.edu

October 5, 2017

Revision Status:

`$Id: FIT2100-Tutorial-06.tex, Version 1.0 2017/10/05 13:30 Jojo $`

Acknowledgement

The majority of the content presented in this tutorial was adapted from:

- William Stallings (2015). *Operating Systems: Internals and Design Principles (8th Edition)*, Pearson.

Contents

1 Background

This tutorial provides students with the opportunity to explore further on the various concepts of memory and I/O managements as discussed in the lectures.

You should complete the suggested reading in Section 2 before attending the tutorial. You should also prepare the solutions for the two sets of practice tasks given in Section 3 and Section 4 respectively.

2 Pre-tutorial Reading

You should complete the following two sets of reading:

- Lecture Notes: Weeks 8, 9, and 10
- Stalling's textbook (7th/8th Edition): Chapters 7, 8, and 11

3 Memory Management

3.1 Review Questions

Question 1

- **Internal fragmentation:** refers to the wasted space internal to a partition due to the fact that the block of data loaded is smaller than the partition.
- **External fragmentation:** is associated with dynamic partitioning, and refers to the fact that a large number of small areas of main memory external to any partitions accumulate.

Question 2

In a paging system, programs and data stored on disk are divided into equal, fixed-sized blocks called **pages**; and the main memory is divided into blocks of the same size called **frames**. One page can fit exactly in one frame.

Question 3

Instead of dividing into fixed-size pages, the program and its associated data are divided into a number of **segments**. It is not required that all segments of all programs be of the same length, although there is a maximum segment length.

Question 4

With virtual memory paging, not all pages of a process need be in main memory frames for the process to run; pages may be read in as needed.

Question 5

- **Demand paging:** a page is brought into main memory only when a reference is made to a location on that page.
- **Prepaging:** pages other than the one demanded by a page fault are brought in.

Question 6

The clock policy is similar to FIFO, except that in the clock policy, any frame with a use bit of 1 is passed over by the algorithm. To some extent, the clock policy keeps track of which frame has just been referenced and protecting it from being replaced.

3.2 Problem-Solving Tasks

3.2.1 Task 1

Refer to page 6.

3.2.2 Task 2

Refer to page 6.

Free Partitions	First-fit	Best-fit	Worst-fit
100K			
500K	(1) Allocate 212K from 500K; Leave a hole of size 288K (3) Allocate 112K from 288K; Leaves a hole of size 176K	(2) Allocate 417K; Leave a hole of 83K	(2) Allocate 417K (83K over allocated)
200K		(3) Allocate 112K; Leave a hole of 88K	
300K		(1) Allocate 212K; Leave a hole of 88K	
600K	(2) Allocate 417K; Leave a hole of size 183K	(4) Allocate 426K; Leave a hole of size 174K	(1) Allocate 212K; Leave a hole of 388K (3) Allocate 112K; Leave a hole of 276K
	(4) Cannot allocate for the demand of 426K		(4) Cannot allocate for the demand of 426K

Figure 1: Section 3.2 Task 1

FIFO	**1**	**2**	**3**	**4**	**5**	**3**	**4**	**1**	**6**	**7**	**8**	**7**	**8**	**9**	**7**	**8**	**9**	**5**	**4**	**5**	**4**	**2**
Frame 1	1	1	1	1	5	5	5	5	5	5	8	8	8	8	8	8	8	8	8	8	8	2
Frame 2		2	2	2	2	2	2	1	1	1	1	1	1	9	9	9	9	9	9	9	9	9
Frame 3			3	3	3	3	3	3	6	6	6	6	6	6	6	6	6	5	5	5	5	5
Frame 4				4	4	4	4	4	4	7	7	7	7	7	7	7	7	7	4	4	4	4

LRU	**1**	**2**	**3**	**4**	**5**	**3**	**4**	**1**	**6**	**7**	**8**	**7**	**8**	**9**	**7**	**8**	**9**	**5**	**4**	**5**	**4**	**2**
Frame 1	1	1	1	1	5	5	5	5	6	6	6	6	6	6	6	6	6	5	5	5	5	5
Frame 2		2	2	2	2	2	2	1	1	1	1	1	1	9	9	9	9	9	9	9	9	9
Frame 3			3	3	3	3	3	3	3	7	7	7	7	7	7	7	7	7	4	4	4	4
Frame 4				4	4	4	4	4	4	4	8	8	8	8	8	8	8	8	8	8	8	2

OPT	**1**	**2**	**3**	**4**	**5**	**3**	**4**	**1**	**6**	**7**	**8**	**7**	**8**	**9**	**7**	**8**	**9**	**5**	**4**	**5**	**4**	**2**
Frame 1	1	1	1	1	1	1	1	1	6	6	8	8	8	8	8	8	8	8	4	4	4	4
Frame 2		2	2	2	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
Frame 3			3	3	3	3	3	3	3	7	7	7	7	7	7	7	7	7	7	7	7	2
Frame 4				4	4	4	4	4	4	4	4	4	4	9	9	9	9	9	9	9	9	9

Figure 2: Section 3.2 Task 2

3.2.3 Task 3

(a) Segment 0 starts at the location 660. With the offset of 198, the physical address is $660 + 198 = 858$

(b) Segment 1 has a length of 422, as such this logical address triggers a segment fault $(515 > 422)$

(c) Segment 3 starts at the location 996. With the offset of 445, the physical address is $996 + 445 = 1441$

4 I/O Management and Disk Scheduling

4.1 Review Questions

Question 1

- **Programmed I/O:** The processor issues an I/O command, on behalf of a process, to an I/O module; the process then *busy-waits* for the operation to be completed before proceeding.
- **Interrupt-driven I/O:** The processor issues an I/O command on behalf of a process, continues to execute subsequent instructions, and is interrupted by the I/O module when the latter has completed its work. The subsequent instructions may be in the same process, if it is not necessary for the process to wait for the completion of the I/O. Otherwise, the process is suspended pending the interrupt and other work (or another process) is performed.
- **Direct memory access (DMA):** A DMA module controls the exchange of data between main memory and an I/O module without the intervention of the processor. The processor sends a request for the transfer of a block of data to the DMA module and is interrupted only after the entire block has been transferred.

Question 2

- **Block-oriented devices:** store information in blocks that are usually of fixed size, and transfers are made one block at a time. Generally, it is possible to reference data by its block number. Disks and tapes are examples of block-oriented devices.
- **Stream-oriented devices:** transfer data in and out as a stream of bytes, with no block structure. Terminals, printers, communications ports, mouse and other pointing devices, and most other devices that are not secondary storage are stream oriented.

Question 3

Double buffering allows two operations to proceed in parallel rather than in sequence. A process can transfer data to (or from) one buffer while the operating system empties (or fills) the other.

Question 4

The delay elements which are involved in a disk read or write include: seek time, rotational delay, and access time.

Question 5

- **FIFO:** Items are processed from the queue in sequential first-come-first-served order.
- **SSTF:** Select the disk I/O request that requires the least movement of the disk arm (head) from its current position.
- **SCAN:** The disk arm moves in one direction only, satisfying all outstanding requests en route, until it reaches the last track in that direction or until there are no more requests in that direction. The service direction is then reversed and the scan proceeds in the opposite direction, again picking up all requests in order.
- **C-SCAN:** Similar to SCAN, but restricts scanning to one direction only. Thus, when the last track has been visited in one direction, the arm is returned to the opposite end of the disk and the scan begins again.

4.2 Problem-Solving Tasks

4.2.1 Task 1

(a) (i) **FIFO:** `27, 129, 110, 186, 147, 41, 10, 64, 120`
(ii) **SSTF:** `110, 120, 129, 147, 186, 64, 41, 27, 10`
(iii) **SCAN:** `64, 41, 27, 10, 110, 120, 129, 147, 186`
(iv) **C-SCAN:** `64, 41, 27, 10, 186, 147, 129, 120, 110`

(b) Average seek length (in terms of the number of tracks traversed):
(i) **FIFO:** `61.8`
(ii) **SSTF:** `29.1`
(iii) **SCAN:** `29.6`
(iv) **C-SCAN:** `38`

4.2.2 Task 2

Each sector (512 bytes) can hold 4 logical records (120 bytes per record). The required number of sectors is `300,000/4 = 75,000` sectors. This requires `75,000/96 = 782` tracks, which in turn requires `782/110 = 8` surfaces.