# Sheet9

## POLICIES FOR VIRTUAL MEMORY

1) Briefly define the alternative page fetch policies.

   With **demand paging**, a page is brought into main memory only when a reference is made to a location on that page. With **prepaging**, pages other than the one demanded by a page fault are brought in.

2) What is the difference between resident set management and page replacement policy?

   **Resident set management** deals with the following two issues: (1) how many page frames are to be allocated to each active process; and (2) whether the set of pages to be considered for replacement should be limited to those of the process that caused the page fault or encompass all the page frames in main memory. **Page replacement policy** deals with the following issue: among the set of pages considered, which particular page should be selected for replacement.

3) What is the relationship between FIFO and clock page replacement algorithms?

   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.

4) Why is it not possible to combine a global replacement policy and a fixed allocation policy?

   Because a fixed allocation policy requires that the number of frames allocated to a process is fixed, when it comes time to bring in a new page for a process, one of the resident pages for that process must be swapped out (to maintain the number of frames allocated at the same amount), which is a local replacement policy.

5) What is the difference between a resident set and a working set?

   The resident set of a process is the current number of pages of that process in main memory. The working set of a process is the number of pages of that process that have been referenced recently.

6) What is the difference between demand cleaning and precleaning?

   With **demand cleaning**, a page is written out to secondary memory only when it has been selected for replacement. A **precleaning** policy writes modified pages before their page frames are needed so that pages can be written out in batches.

7) Consider the following string of page references 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2. Complete a figure similar to Figure 8.14, showing the frame allocation for:

   a) FIFO (first-in-first-out)

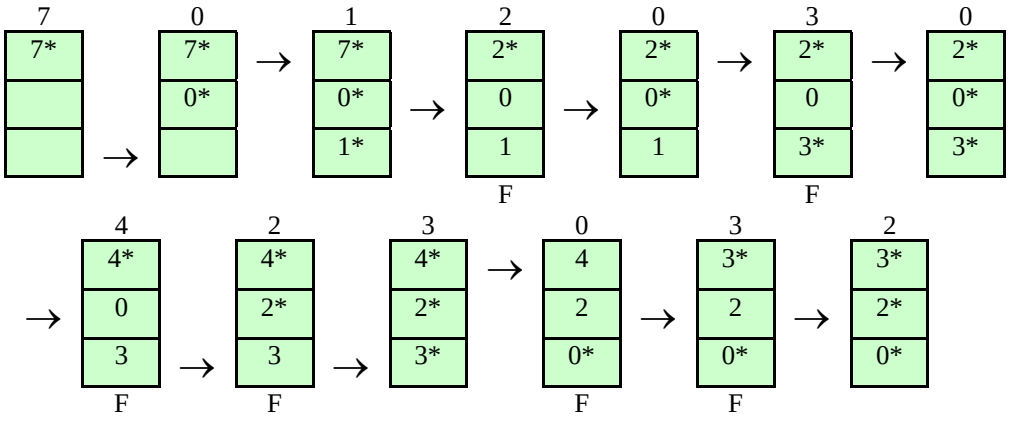
| 7 | 0 | 1 | 2 | 0 | 3 | 0 | 4 | 2 | 3 | 0 | 3 | 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 7 | 7 | 7 | 2 | 2 | 2 | 2 | 4 | 4 | 4 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 2 |
| | | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 3 | 3 | 3 | 3 |
| | | | F | | F | F | F | F | F | F | | |

   b) LRU (least recently used)

| 7 | 0 | 1 | 2 | 0 | 3 | 0 | 4 | 2 | 3 | 0 | 3 | 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 7 | 7 | 7 | 2 | 2 | 2 | 2 | 4 | 4 | 4 | 0 | 0 | 0 |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 3 | 3 |
| | | 1 | 1 | 1 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 2 |
| | | | F | | F | | F | F | F | F | | |

c) Clock




d) Optimal (assume the page reference string continues with 1, 2, 0, 1, 7, 0, 1)

| 7 | 0 | 1 | 2 | 0 | 3 | 0 | 4 | 2 | 3 | 0 | 3 | 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 7 | 7 | 7 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 4 | 4 | 0 | 0 | 0 |
| | | 1 | 1 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| | | | F | | F | | | | | F | | |

e) List the total number of page faults and the miss rate for each policy. Count page faults only after all frames have been initialized.

FIFO: page faults = 7 miss rate = 70%
LRU: page faults = 6 miss rate = 60%
Clock: page faults = 6 miss rate = 60%
OPT: page faults = 3 miss rate = 30%

8) A process references five pages, A, B, C, D, and E, in the following order:

A; B; C; D; A; B; E; A; B; C; D; E

Assume that the replacement algorithm is first-in-first-out and find the number of page transfers during this sequence of references starting with an empty main memory with three page frames. Repeat for four page frames.

9 and 10 page transfers, respectively. This is referred to as "Belady's anomaly," and was reported in "An Anomaly in Space-Time Characteristics of Certain Programs Running in a Paging Machine," by Belady et al, Communications of the ACM, June 1969.

9) A process contains eight virtual pages on disk and is assigned a fixed allocation of four page frames in main memory. The following page trace occurs:

1, 0, 2, 2, 1, 7, 6, 7, 0, 1, 2, 0, 3, 0, 4, 5, 1, 5, 2, 4, 5, 6, 7, 6, 7, 2, 4, 2, 7, 3, 3, 2, 3

a) Show the successive pages residing in the four frames using the LRU replacement policy. Compute the hit ratio in main memory. Assume that the frames are initially empty.

Hit ratio = 16/33

| 1 | 0 | 2 | 2 | 1 | 7 | 6 | 7 | 0 | 1 | 2 | 0 | 3 | 0 | 4 | 5 | 1 | 5 | 2 | 4 | 5 | 6 | 7 | 6 | 7 | 2 | 4 | 2 | 7 | 3 | 3 | 2 | 3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| - | 0 | 0 | 0 | 0 | 0 | 6 | 6 | 6 | 6 | 2 | 2 | 2 | 2 | 2 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| - | - | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| - | - | - | - | - | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 3 | 3 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 3 | 3 | 3 | 3 |
| F | F | F | | | F | F | | F | | F | | F | | F | F | F | | F | | | F | F | | | F | F | | | F | | | |

b) Repeat part (a) for the FIFO replacement policy.

Hit ratio = 16/33

| 1 | 0 | 2 | 2 | 1 | 7 | 6 | 7 | 0 | 1 | 2 | 0 | 3 | 0 | 4 | 5 | 1 | 5 | 2 | 4 | 5 | 6 | 7 | 6 | 7 | 2 | 4 | 2 | 7 | 3 | 3 | 2 | 3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 2 | 2 |
| - | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| - | - | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| - | - | - | - | - | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 |
| F | F | F | | | F | F | | | F | | F | F | | F | F | F | | F | | | F | F | | | | F | | | F | | F | |

c) Compare the two hit ratios and comment on the effectiveness of using FIFO to approximate LRU with respect to this particular trace.

These two policies are equally effective for this particular page trace.

10) Consider a page reference string for a process with a working set of M frames, initially all empty. The page reference string is of length P with N distinct page numbers in it. For any page replacement algorithm,

a) What is a lower bound on the number of page faults?

N

b) What is an upper bound on the number of page faults?

P

---