Operating Systems
Jacobs University Bremen
Dr. Jürgen Schönwälder

Course: CO20-320202
Date: 2018-10-30
Due: 2018-11-08

OS 2018 Problem Sheet #4

Problem 4.1: *address spaces in a paging system* (2 points)

Consider a operating system that uses paging for memory management with a page size of 1024 bytes. The logical address space of processes is limited to a maximum of 64 pages. The physical memory has a size of 256 kilobytes.

a) How many frames has the physical memory?

b) How many bits has an address in the logical address space?

c) How many bits has an address in the physical address space?

d) How many bits are used for the page number?

e) How many bits are used for the offset within a page?

Problem 4.2: *page replacement algorithms* (1+1+1+1+1+1 = 6 points)

An operating system uses demand paging to implement virtual memory. A user-space process uses four pages, numbered 1 to 4. The pages are accessed in the following order (reference string): 1, 2, 3, 4, 1, 1, 4, 2, 1, 2. None of the pages used by the user-space process are resident when the process starts and the frames allocated to the process are unused. Fill out the following tables by writing page numbers into the cells. Each number indicates which page a frame holds after accessing the page indicated by the reference string. Write down the number of page faults for each scenario.

Hint: Do not move pages between frames except when necessary.

a) Show how the pages are mapped to two frames using the First-In-First-Out (FIFO) page replacement algorithm.

reference string	1	2	3	4	1	1	4	2	1	2
frame 0										
frame 1										

b) Show how the pages are mapped to three frames using the First-In-First-Out (FIFO) page replacement algorithm.

reference string	1	2	3	4	1	1	4	2	1	2
frame 0										
frame 1										
frame 2										

c) Show how the pages are mapped to two frames using Belady's Optimal (BO) page replacement algorithm.

reference string	1	2	3	4	1	1	4	2	1	2
frame 0										
frame 1										

d) Show how the pages are mapped to three frames using Belady's Optimal (BO) page replacement algorithm.

reference string	1	2	3	4	1	1	4	2	1	2
frame 0										
frame 1										
frame 2										

e) Show how the pages are mapped to two frames using the Least Recently Used (LRU) page replacement algorithm.

reference string	1	2	3	4	1	1	4	2	1	2
frame 0										
frame 1										

f) Show how the pages are mapped to three frames using the Least Recently Used (LRU) page replacement algorithm.

reference string	1	2	3	4	1	1	4	2	1	2
frame 0										
frame 1										
frame 2										

Problem 4.3: *working set vs least recently used* (2 points)

Consider the design of a paging system that tracks the working set of all processes and aims at keeping the working sets of the processes resident. If pages are removed from the working set of a process, then these pages are removed from the resident set of pages and the frames get marked as unused. In other words, only pages that are part of the working set are resident, all other pages are not.

Assume the working set is calculated over the last N memory accesses for every process. Compare this system with a system that uses the least recently used (LRU) page replacement strategy for each process with N pages allocated to each process. What can you say about the pages that are resident in both systems at any point in time?

Which system do you think is preferable? Can you suggest further improvements?