Operating Systems
Constructor University
Dr. Jürgen Schönwälder

Module: CO-562
Date: 2023-10-20
Due: 2023-10-27

Problem Sheet #7

Problem 7.1: *positioning algorithms* (1+1+1+1 = 4 points)

A system using segmentation has free blocks of the following sizes:

```
16 KiB,  8 KiB,  6 KiB, 21 KiB, 14 KiB, 10 KiB
```

The blocks are ordered as shown above and the last memory allocation took place right before the first block. The system receives the following sequence of memory requests (in the order shown below):

```
9 KiB,  11 KiB,  7 KiB, 16 KiB
```

If a free block is allocated that is bigger than the request, then a new small block is left over and positioned at the same place where the bigger block was positioned in the list of free blocks.

a) Show how the list of free blocks changes after each allocation using the best-fit algorithm.

Request	16 KiB	8 KiB	6 KiB	21 KiB	14 KiB	10 KiB
9 KiB						
11 KiB						
7 KiB						
16 KiB						

b) Show how the list of free blocks changes after each allocation using the worst-fit algorithm.

Request	16 KiB	8 KiB	6 KiB	21 KiB	14 KiB	10 KiB
9 KiB						
11 KiB						
7 KiB						
16 KiB						

c) Show how the list of free blocks changes after each allocation using the first-fit algorithm.

Request	16 KiB	8 KiB	6 KiB	21 KiB	14 KiB	10 KiB
9 KiB						
11 KiB						
7 KiB						
16 KiB						

d) Show how the list of free blocks changes after each allocation using the next-fit algorithm.

Request	16 KiB	8 KiB	6 KiB	21 KiB	14 KiB	10 KiB
9 KiB						
11 KiB						
7 KiB						
16 KiB						

Problem 7.2: *buddy system* (1+1+1 = 3 points)

A system is using a buddy system memory allocator and it has initially $1024\,\mathrm{KiB}$ of free memory available. The memory allocator receives the following sequence of memory allocations:

A: $210\,\mathrm{KiB}$, B: $20\,\mathrm{KiB}$, C: $180\,\mathrm{KiB}$, D: $100\,\mathrm{KiB}$, E: $90\,\mathrm{KiB}$, F: $60\,\mathrm{KiB}$, G: $50\,\mathrm{KiB}$, H: $40\,\mathrm{KiB}$

a) Draw a diagram showing how the allocations are placed in the free memory. Draw also the associated binary tree.

b) Calculate the overall internal fragmentation. What is the largest chunk of memory that can still be allocated?

c) Allocation F is deallocated and afterwards an allocation request I: $80\,\mathrm{KiB}$ is received. Can this allocation be accomodated? Explain why or why not.

Problem 7.3: *page replacement algorithms* (1+1+1 = 3 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): 3, 1, 2, 4, 1, 3, 2, 4, 1, 3. 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.

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

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

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

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

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

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

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

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

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

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

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

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

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