# Modern Operating System Exercise 4

UNIkeEN

March 16, 2023

## Problem 1

Assume the page size is $k$ bytes. So we have $\frac{256M}{k}$ virtual page numbers.The page table has $\frac{256M}{k}$ entries, entry size is 4 bytes, the total size is $4 \times \frac{256M}{k}$ bytes.

The entire table needs to fit well in one page, so we have $4 \times \frac{256M}{k} \leq k$, the minimum page size is 32KB.

## Problem 2

The allocation situation is shown in the figure below. The allocation order of processes is the same as their arrival order.

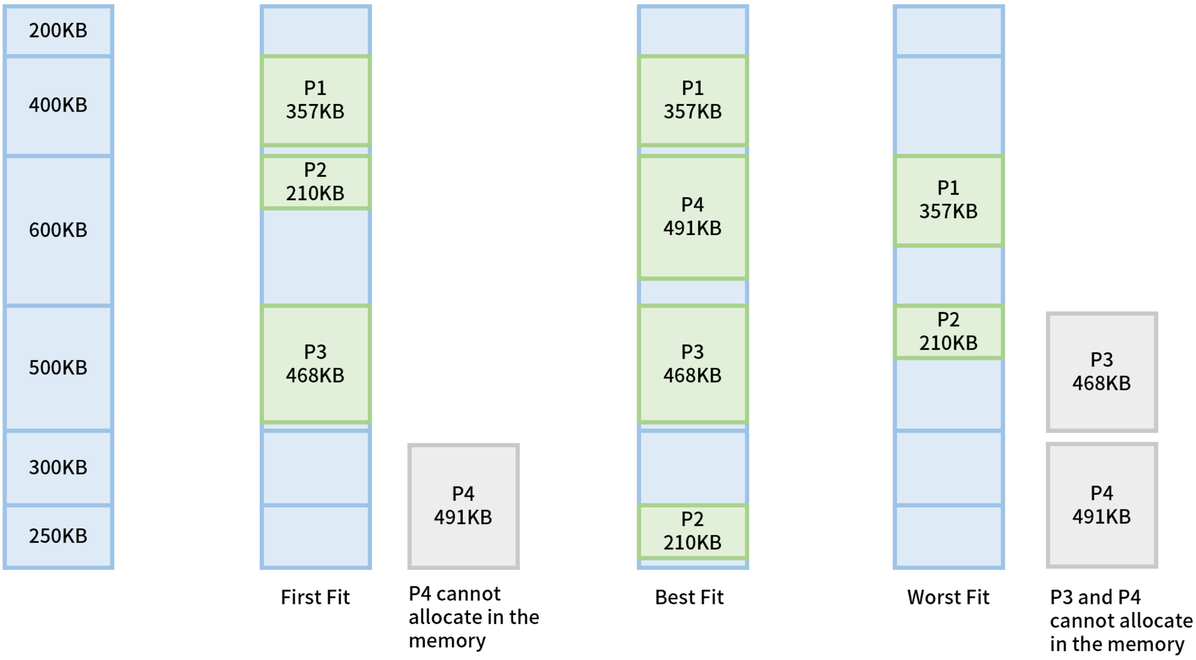

**Fig. 1.** The allocation situation

Using First-Fit algorithm, P4 cannot allocate in the memory, while using Worst-Fit algorithm, P3 and P4 cannot allocate in the memory.

# Problem 3

1. FIFO page replacement algorithm

| Ref String | 1 | 2 | 3 | 4 | 2 | 1 | 5 | 6 | 2 | 1 | 2 | 3 | 7 | 6 | 3 | 2 | 1 | 2 | 3 | 6 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 1 | 1 | 1 | 1 | 1 | 5 | 5 | 5 | 5 | 5 | 3 | 3 | 3 | 3 | 3 | 1 | 1 | 1 | 1 |
| | | 2 | 2 | 2 | 2 | 2 | 2 | 6 | 6 | 6 | 6 | 6 | 7 | 7 | 7 | 7 | 7 | 7 | 3 | 3 |
| | | | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
| | | | | 4 | 4 | 4 | 4 | 4 | 4 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 |
| Hit/Miss | M | M | M | M | H | H | M | M | M | M | H | M | M | M | H | M | M | H | M | H |

Page Faults: 14, Hit Ratio: 0.3

2. LRU page replacement algorithm

| Ref String | 1 | 2 | 3 | 4 | 2 | 1 | 5 | 6 | 2 | 1 | 2 | 3 | 7 | 6 | 3 | 2 | 1 | 2 | 3 | 6 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
| | | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| | | | 3 | 3 | 3 | 3 | 5 | 5 | 5 | 5 | 5 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| | | | | 4 | 4 | 4 | 4 | 6 | 6 | 6 | 6 | 6 | 7 | 7 | 7 | 7 | 1 | 1 | 1 | 1 |
| Hit/Miss | M | M | M | M | H | H | M | M | H | H | H | M | M | M | H | H | M | H | H | H |

Page Faults: 10, Hit Ratio: 0.5

3. Second-chance page replacement algorithm

| Ref String | 1 | 2 | 3 | 4 | 2 | 1 | 5 | 6 | 2 | 1 | 2 | 3 | 7 | 6 | 3 | 2 | 1 | 2 | 3 | 6 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 1 | 1 | 1 | 1 | 1 | 5 | 5 | 5 | 5 | 5 | 3 | 3 | 3 | 3 | 3 | 1 | 1 | 1 | 1 |
| | | 2 | 2 | 2 | 2 | 2 | 2 | 6 | 6 | 6 | 6 | 6 | 7 | 7 | 7 | 7 | 7 | 7 | 3 | 3 |
| | | | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
| | | | | 4 | 4 | 4 | 4 | 4 | 4 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 |
| Hit/Miss | M | M | M | M | H | H | M | M | M | M | H | M | M | M | H | M | M | H | M | H |

Page Faults: 14, Hit Ratio: 0.3

4. OPT page replacement algorithm

Note: The question did not ask for this algorithm. The situation of OPT algorithm was calculated by reading the wrong question before, and it is reserved here.

| Ref String | 1 | 2 | 3 | 4 | 2 | 1 | 5 | 6 | 2 | 1 | 2 | 3 | 7 | 6 | 3 | 2 | 1 | 2 | 3 | 6 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 7 | 7 | 7 | 7 | 1 | 1 | 1 | 1 |
| | | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| | | | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| | | | | 4 | 4 | 4 | 5 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
| Hit/Miss | M | M | M | M | H | H | M | M | H | H | H | H | M | H | H | H | M | H | H | H |

Page Faults: 8, Hit Ratio: 0.6

# Problem 4

1. Page size is 8KB, page offset address is $\log 8K$=13 digits. Page number part is 46-13=33 digits. Each page can contain $\frac{8K}{4} = 2^{11}$ page entries.(Up to 11 digit page numbers can be represented) So we need $\frac{33}{11} = 3$ levels.
2. 4 memory operations are required (3 for page tables, 1 for the target byte)
3. At least, we need 6 pages. (3 for code,data and stack page, and 3 for three levels of page tables).

   However, Considering that both the code page and the data page are located at a low address and the stack page is located at a high address, it may be necessary to have different secondary and tertiary page tables pointing to the stack page and the other two pages, so **a total of 8 pages are needed**.(3 for code,data and stack page, 1 for the first page table, 2×2 for the secondary and tertiary page tables)