

Homework 09

CS307-Operating System (D), Chentao Wu, Spring 2020.

Name: 方泓杰(Hongjie Fang) Student ID: 518030910150 Email: galaxies@sjtu.edu.cn

- (9.6) Given six memory partitions of 300 KB, 600 KB, 350 KB, 200 KB, 750 KB and 125 KB (in order), how would the first-fit, best-fit and worst-fit algorithms place processes of size 115 KB, 500 KB, 358 KB, 200 KB, and 375 KB (in order)?

Solution. Here are the situations of three algorithms.

- **(First-fit Algorithm)** Here are the situations displayed in order.
 - i. 115 KB will be put in the 300 KB memory partition, then the rest of the memory partitions are (185 KB, 600 KB, 350 KB, 200 KB, 750 KB, 125 KB);
 - ii. 500 KB will be put in the 600 KB memory partition, then the rest of the memory partitions are (185 KB, 100 KB, 350 KB, 200 KB, 750 KB, 125 KB);
 - iii. 358 KB will be put in the 750 KB memory partition, then the rest of the memory partitions are (185 KB, 100 KB, 350 KB, 200 KB, 392 KB, 125 KB);
 - iv. 200 KB will be put in the 350 KB memory partition, then the rest of the memory partitions are (185 KB, 100 KB, 150 KB, 200 KB, 392 KB, 125 KB);
 - v. 375 KB will be put in the 392 KB memory partition, then the rest of the memory partitions are (185 KB, 100 KB, 150 KB, 200 KB, 17 KB, 125 KB).
- **(Best-fit Algorithm)** Here are the situations displayed in order.
 - i. 115 KB will be put in the 125 KB memory partition, then the rest of the memory partitions are (300 KB, 600 KB, 350 KB, 200 KB, 750 KB, 10 KB);
 - ii. 500 KB will be put in the 600 KB memory partition, then the rest of the memory partitions are (300 KB, 100 KB, 350 KB, 200 KB, 750 KB, 10 KB);
 - iii. 358 KB will be put in the 750 KB memory partition, then the rest of the memory partitions are (300 KB, 100 KB, 350 KB, 200 KB, 392 KB, 10 KB);
 - iv. 200 KB will be put in the 200 KB memory partition, then the rest of the memory partitions are (300 KB, 100 KB, 350 KB, 0 KB, 392 KB, 10 KB);
 - v. 375 KB will be put in the 392 KB memory partition, then the rest of the memory partitions are (300 KB, 100 KB, 350 KB, 0 KB, 17 KB, 10 KB).
- **(Worst-fit Algorithm)** Here are the situations displayed in order.
 - i. 115 KB will be put in the 750 KB memory partition, then the rest of the memory partitions are (300 KB, 600 KB, 350 KB, 200 KB, 635 KB, 125 KB);
 - ii. 500 KB will be put in the 635 KB memory partition, then the rest of the memory partitions are (300 KB, 600 KB, 350 KB, 200 KB, 135 KB, 125 KB);
 - iii. 358 KB will be put in the 600 KB memory partition, then the rest of the memory partitions are (300 KB, 242 KB, 350 KB, 200 KB, 135 KB, 125 KB);
 - iv. 200 KB will be put in the 350 KB memory partition, then the rest of the memory partitions are (300 KB, 242 KB, 150 KB, 200 KB, 135 KB, 125 KB);
 - v. 375 KB can not be put in any of the current memory partitions, since it is larger than any partition. Therefore, the process of size 375 KB must wait until there is a memory partition whose capacity is larger than its size.

□

- (9.7) Assuming a 1-KB page size, what are the page numbers and offsets for the following address references (provided as decimal numbers):
 - a. 3085;
 - b. 42095;
 - c. 215201;
 - d. 650000;
 - e. 2000001.

Solution. The page size is 1 KB, which means the offsets have 10 binary bits. Therefore, we can derive the following address references by transforming them into binary mode.

- a. $(3085)_{10} = (1100\ 0000\ 1101)_2$; hence,

$$\begin{aligned}\text{offset} &= (00\ 0000\ 1101)_2 = 13, \\ \text{page number} &= (11)_2 = 3.\end{aligned}$$

- b. $(42095)_{10} = (1010\ 0100\ 0110\ 1111)_2$; hence,

$$\begin{aligned}\text{offset} &= (00\ 0110\ 1111)_2 = 111, \\ \text{page number} &= (10\ 1001)_2 = 41.\end{aligned}$$

- c. $(215201)_{10} = (0011\ 0100\ 1000\ 1010\ 0001)_2$; hence,

$$\begin{aligned}\text{offset} &= (00\ 1010\ 0001)_2 = 161, \\ \text{page number} &= (00\ 1101\ 0010)_2 = 210.\end{aligned}$$

- d. $(650000)_{10} = (1001\ 1110\ 1011\ 0001\ 0000)_2$; hence,

$$\begin{aligned}\text{offset} &= (11\ 0001\ 0000)_2 = 784, \\ \text{page number} &= (10\ 0111\ 1010)_2 = 634.\end{aligned}$$

- e. $(2000001)_{10} = (0001\ 1110\ 1000\ 0100\ 1000\ 0001)_2$; hence,

$$\begin{aligned}\text{offset} &= (00\ 1000\ 0001)_2 = 129, \\ \text{page number} &= (00\ 0111\ 1010\ 0001)_2 = 1953.\end{aligned}$$

□

- (9.9) Consider a logical address space of 256 pages with a 4-KB page size, mapped onto a physical memory of 64 frames.
 - a. How many bits are required in the logical address?
 - b. How many bits are required in the physical address?

Solution. The 4-KB page size means an offset of 12 binary bits.

- a. The logical address space of 256 pages, which means a page number of 8 binary bits. Therefore, $12 + 8 = 20$ bits are required in the logical address.

- b. The physical address space of 64 frames, which means a frame number of 6 binary bits. Therefore, total $12 + 6 = 18$ bits are required in the physical address.

□

- (9.10) Consider a computer system with a 32-bit logical address and 4-KB page size. The system supports up to 512 MB of physical memory. How many entries are there in each of the following?
 - a. A conventional, single-level page table;
 - b. An inverted page table.

Solution. The 4-KB page size means an offset of 12 binary bits. 512 MB of physical memory means the physical address of 29 binary bits. Therefore,

- A page number is represented by $32 - 12 = 20$ binary bits, which means the logical address space of 2^{20} pages;
- A frame number is represented by $29 - 12 = 17$ binary bits, which means the physical address space of 2^{17} frames.

Hence, we can derive the number of entries in each situation.

- a. **(A conventional, single-level page table)** The number of entries in a conventional, single-level page table should equal to the total page number of the logical address space, which is $2^{20} = 1\text{M} = 1048576$;
- b. **(An inverted page table)** The number of entries in an inverted page table should equal to the total frame number of the physical address space, which is $2^{17} = 128\text{K} = 131072$.

□