Stamatios Morellas
COM S 352 - Homework 8
Due 4/5/20 @ 11:59pm

(20 Points) Exercise 9.13

Let `MP[] = { 100MB, 170MB, 40MB, 205MB, 300MB, 185MB }`
Let `S[] = { 200MB, 15MB, 185MB, 75MB, 175MB, 80MB }`

First-fit:

- S[0] occupies 200MB of MP[3], leaving 5MB free in that partition
- S[1] occupies 15MB of MP[0], leaving 75MB free in that partition
- S[2] occupies 185MB of MP[4], leaving 115MB free in that partition
- S[3] occupies 75MB of MP[0], leaving 10MB free in that partition
- S[4] must wait since there is not enough space to accommodate this process
- S[5] occupies 80MB of MP[1], leaving 90MB free in that partition

Best-fit:

- S[0] occupies 200MB of MP[3], leaving 5MB free in that partition
- S[1] occupies 15MB of MP[2], leaving 25MB free in that partition
- S[2] occupies 185MB of MP[4], leaving 115MB free in that partition
- S[3] occupies 75MB of MP[0], leaving 25MB free in that partition
- S[4] occupies 175MB of MP[5], leaving 10MB free in that partition
- S[5] occupies 80MB of MP[1], leaving 90MB free in that partition

Worst-fit:

- S[0] occupies 200MB of MP[4], leaving 100MB free in that partition
- S[1] occupies 15MB of MP[3], leaving 190MB free in that partition
- S[2] occupies 185MB of MP[3], leaving 5MB free in that partition
- S[3] occupies 75MB of MP[5], leaving 110MB free in that partition
- S[4] has to wait since there is not enough memory to allocate space for
- S[5] occupies 80MB of MP[0], leaving 20MB free in that partition

(10 Points) Exercise 9.15

External fragmentation causes problems in terms of the contiguous memory allocation scheme since holes will develop as a result of new processes being initiated after the death of old ones. The pure segmentation scheme also suffers from external fragmentation, but pure paging is the only scheme that does not suffer from external fragmentation. It does, however, suffer from internal fragmentation, which involves the wasting of space in memory if a page is not fully utilized.

(12 Points) Exercise 9.21

page number = address reference / page size
offset = address reference % page size

a) 21205
→ binary = 00000101001011010101
→ page num = 10100 = 20
→ page offset = 1011010101 = 725
b) 164250
→ binary = 00101000000110011010
→ page num = 0010100000 = 160
→ page offset = 0110011010 = 410

(12 points) A process' logical address consists of 4 pages and page size is 2KB. The page table of the process is given in Figure 9.9. Compute the physical address for each of the following logical addresses (provided as decimal numbers).

a) 1500
→ number = 1500 / 2096 = 0
→ offset = 1500 % 2096 = 1500
→ frame for page 0 = 1
→ physical address = **3596**
b) 7500
→ number = 7500 / 2096 = 3
→ offset = 7500 % 2096 = 1212
→ frame for page 3 = 7
→ physical address = **15884**

(10 points) Exercise 9.23

Pages = 2,048 = 2^{11} pages
Page size = 4-KB = 2^{12} bytes
Frames = 512 = 2^{23} frames

Bits required in logical address = ($2^{11} * 2^{12}$) = 2^{23} bytes = **23 bits**
Bits required in physical address = ($2^{9} * 2^{12}$) = 2^{21} bytes = **21 bits**

(6 points) Consider a computer system with a 32-bit logical address and 8-KB page size. The system supports up to 1GB of physical memory. How many entries are there in a page table?

Logical memory spaces = 2^{32} bytes
Page size = 8-KB = 2^{13} bytes
Number of pages = $2^{32} / 2^{13} = 2^{19}$
Entries in a conventional single-level page table = 2^{19}