Matt Ritchie

CS420

HW6

1) Given five memory partitions of 100 KB, 500 KB, 200 KB, 300 KB, and

600 KB (in order), how would each of the first-fit, best-fit, and worst-fit

algorithms place processes of 212 KB, 417 KB, 112 KB, and 426 KB (in order)?

Which algorithm makes the most efficient use of memory?

First-fit algorithm:

212K -> 500K partition

417K -> 600K partition

112K -> 288K partition (make new partition 288K = 500K - 212K)

426K must wait

Best-fit algorithm:

212K -> 300K partition

417K -> 500K partition

112K -> 200K partition

426K -> 600K partition

Worst-fit algorithm:

212K -> 600K partition

417K -> 500K partition

112K -> 388K partition

426K must wait

So best-fit works best here because it’s the only one that doesn’t have anything waiting.

2) Assuming a 1 KB (1024 bytes) page size, what are the page numbers and offsets for the following address references (provided as decimal numbers). Assume that both pages and page offsets are indexed starting at 0. That is, the first page is page 0.

a) 2375

b) 19366

c) 30000

d) 256

e) 16385

a. page = 1; offset = 327

b. page = 18; offset = 934

c. page = 29; offset = 304

d. page = 0; offset = 256

e. page = 16; offset = 1

3) Consider a logical address space of 32 pages with 1024 bytes per page;

mapped onto a physical memory of 16 frames.

1. How many bits are required in the logical address?

16bits

1. How many bits are required in the physical address?

15bits

4) Consider a paging system with the page table stored in memory.

a) If a memory reference takes 200 nanoseconds, how long does a paged

memory reference take?

400 nanoseconds: 200 nanoseconds to access the page table and 200

nanoseconds to access the memory

b) If we add associative registers, and 75 percent of all page-table

references are found in the associative registers, what is the effective memory reference time? (Assume that finding a page-table entry in the associative registers takes zero time if the entry is there.)

Effective access time = 0.75 × (200 nanoseconds) + 0.25 × (400

nanoseconds) = 250 nanoseconds