Operating Systems – Exercise 3

Scheduling & Cache

Don’t forget – According to the course regulations, **your allowed to submit an “I don’t know” PDF** file. Doing so will automatically grant you a grade of 50 on the assignment.

Due date: **15/6/23, 23:55**

Question 1 (25 points)

Five batch jobs, A, B, C, D, E, arrive at a computer center at almost the same time. They have estimated running times of 11, 6, 2, 4, and 8 minutes.
For each of the following scheduling algorithms, determine the mean process turnaround time for each method:

(a) Round-robin

(b) First come, First served (run in order 11, 6, 2, 4, 8)

(c) Shortest job first

Ignore process switching overhead.

For (a), assume that the system is multiprogrammed, and that each job gets its fair share of the CPU. For (b) and (c) assume that only one job runs at a time, until it finishes. All jobs are completely CPU bound. Assume that the time quantum of the scheduler is 1 minute.

Question 2 (25 points)

Answer with True/False and explain.

(2.1) The decision of what is being kept in L1 cache decided by the OS

True/False

(2.2) If we change the implementation of a direct-mapped cache to a fully associative cache, then the hit rate will rise (in an experiment over time for the average use case).

True/False

(2.3) A fully-associative cache has a higher space overhead than a direct-mapped cache.

True/False

Question 3 (25 points):

For each of the following sections regarding cache memory - what is the amount of bits assigned for index, tag, and offset?

Specify details for 3 different mapping mechanisms, and explain your answer.

1. Direct Mapping
2. 4-way Set associativity
3. Fully associative cache

a. Our computer uses a cache memory with 4096 blocks. Each block in the cache memory consists of 8 words. Word size is 32 bits (4 Bytes).
   Address size is 64 bit.

b. Our computer uses a cache memory with 1024 blocks. Each block in the cache memory consists of 4 words. Word size is 16 bits (2 Bytes).
   Address size is 16 bit.

<u>Question 4 (25 points):</u>

Assume a computer has 32 bit addresses with **word** addressable mechanism. Each block stores 16 words. A direct-mapped cache has 256 blocks. In which block (index) of the cache would we look for each of the following addresses?

Addresses are given in hexadecimal for convenience.

a. B4E341B1
b. C2E87796
c. D6312D0F
d. 16FF913C