DAT 103
Datamaskiner og operativsystemer (Computers and Operating Systems)

Supplementary exercises (Set 6)

Problem 1

Assume the size of a page is 4096 bytes. If a process requests 57580 bytes,

(a) How many pages will the process be allocated?

Solution.
$57580/4096 = 14$ pages $+ 236$ bytes $= 15$ pages

(b) What is the size of the internal fragmentation for this memory allocation?

Solution.
$4096 - 236 = 3860$ bytes

Problem 2

What are the differences between logical and physical addresses?

Problem 3

Assuming a $1 - KB$ page size, what are the page numbers and offsets for the following address references (provided as decimal numbers):

(a) 3085

Solution.
page = 3; offset = 13

(b) 215201

Solution.
page = 210; offset = 161

Problem 4

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 a conventional, single-level page table? **Solution.**
2^{20} entries.

Problem 5

Explain the difference between internal and external fragmentation.

Problem 6

Consider a processor that needs 10 ns to access the cache memory and 100 ns to access the main memory. Assume the cache hit ratio is 90%, what is the average memory access time of the processor? **Solution.**
10*0.9+(10+100)*0.1=20ns

Problem 7

Consider a paging hardware with a translation look-aside buffer (TLB). Assume that the page table and *all the pages* a process requests are in the physical memory. Suppose the percentage of times that the page number of interest is found in the TLB[1] is 80%, each TLB lookup takes 10 ns, and each memory access takes 90 ns. What is the effective access time (EAT)?

Solution.
$0.8 \times (10 + 90) + 0.2 \times (10 + 90 + 90) = 118$ ns

- $0.8 \times (10 + 90)$ refers to the case that the page number we need is found in TLB, and only one memory access is needed.
- $0.2 \times (10 + 90 + 90)$ refers to the case that the page number we need is NOT found in TLB, and we need to go to the page table in main memory to find the page number. Thus, two memory accesses are needed.

Problem 8

Consider a paging hardware with a translation look-aside buffer (TLB). Suppose

- each TLB lookup takes 5 ns,
- updating an entry in TLB takes 10 ns,
- each memory access takes 100 ns, and
- moving one page in/out of the memory from/to the backing store takes 500 000 ns.

Consider a process having the page table and **some** of its requested pages in the physical memory. Assume that the process tries to access a page. Calculate the total time required to access this page for each of the following scenario. Note that you need to show how you derive the answer.

8.1 TLB hit for the page entry without page fault

Solution.
5 + 100 = 105 ns

8.2 TLB miss for the page entry without page fault

Solution.
5 + 2 × 100 + 10 = 215 ns

8.3 TLB miss for the page entry with page fault and there is free frame in the main memory

Solution.

PT: page table

TLB lookup + PT + backing store + update PT + update TLB
5 + 100 + 500 000 + 100 + 10 = 500 215 ns ;
or TLB lookup + PT + backing store + update PT + access the newly fetched page + update TLB
5 + 100 + 500 000 + 100 + 100 + 100 + 10 = 500 315 ns

(***Hint****: The TLB needs to be updated in the case of TLB miss, while the page table has to be updated if there is a page fault*).

[1]It is also called TLB hit ratio