https://folk.ntnu.no/michaeng/tdt4186_22/
michael.engel@ntnu.no

Theoretical exercises
Spring 2022

Theoretical Exercises 4

Memory allocation and virtual memory

Please submit solutions on Blackboard by Monday, 28.2.2022 12:00h

4.1 Replacement strategies

Perform and visualize (as shown in lecture 10) the access sequence with the replacement strategies FIFO, Optimal and LRU once with a memory with a capacity of 4 pages and once with 5 pages. Calculate the "hit rate" (accesses which did not result in a replacement operation) for all scenarios.

Request sequence: 1, 3, 5, 4, 2, 4, 3, 2, 1, 0, 5, 3, 5, 0, 4, 3, 5, 4, 3, 2, 1, 3, 4, 5

4.2 More replacement strategies

A computer has four page frames. The time of loading, time of last access, and the R (reference) and M (modified) bits for each page are as shown below (the times are in clock ticks):

Page	Loaded	Last ref.	R	M
0	126	280	1	0
1	230	265	0	1
2	140	270	0	0
3	110	285	1	1

Which pages will the algorithms FIFO, LRU and Second Chance replace? Explain your answer!

4.3 Buddy allocation

The Buddy method for allocating memory to processes shall be used for a memory with a capacity of 1024 kB. Perform the provided operations and give the occupancy state of the memory after each operation. Indicate if an allocation cannot be satisfied.

a. Request 65 kB (A)

b. Request 30 kB (B)

c. Request 90 kB (C)

d. Request 34 kB (D)

e. Request 130 kB (E)

f. Release C

g. Release B

h. Request 275 kB (F)

i. Request 145 kB (G)

j. Release D

k. Release A

l. Release G

m. Release E

4.4 Virtual memory

A machine has a physical memory with 2^{32} addressable bytes and a page size of 8 KB. Each process is allocated a virtual address space of 4 GB. Page table entries are 32 bits long. Page tables are kept in pageable memory.

a. Why is one-level paging inadequate for this system?

b. Why is two-level paging sufficient?

c. How many bits are needed to reference the outer page table and how many to reference the inner page table? Explain your answer showing all appropriate arithmetic.

4.5 Paging and memory accesses

Consider the following two-dimensional array (assume `sizeof(int) = 8`):

```
int X[32][32];
```

Suppose that a system uses 4 pages of 512 byte page size each.

The X array is stored in row-major order (i.e., `X[0][1]` follows `X[0][0]` in memory). Which of the following two code fragments will generate the lower number of page faults?

Compute the total number of page faults for each code fragment. Explain your calculation.

Fragment 1:

```
for (int j=0; j<32; j++)
  for (int i=0; i<32; i++)
    X[i][j]++;
```

Fragment 2:

```
for (int i=0; i<32; i++)
  for (int j=0; j<32; j++)
    X[i][j]++;
```