Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich
Spring Term 2014

Operating Systems and Networks
Assignment 4

Assigned on: **13th March 2014**
Due by: **21st March 2014**

1 Memory Management

What are the goals of memory managent in a modern OS?

- Provide memory to applications as a contiguous address space.
- Protect the memory of one process from access by other processes, but allow sharing memory under some circumstances.
- Provide more virtual memory than physical memory available.

1.1 Segmentation

Consider the following segment table:

Segment	Base	Length
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

What are the physical addresses for the following logical addresses given as (segment, offset) tuples?

a) 0, 430: 219 + 430 = 649

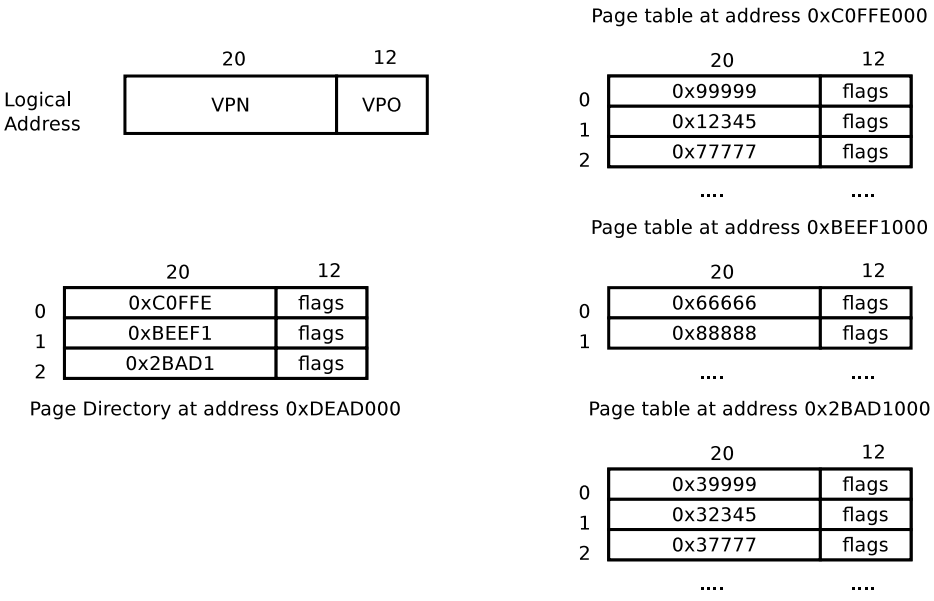
b) 1, 10: 2300 + 10 = 2310

c) 2, 500: illegal reference

d) 3, 400: 1327 + 400 = 1727

e) 7, 112: illegal reference

1.2 Paging



Answer the following questions concerning the given P6 page table:

a) How does paging provide isolation of processes?

b) How does paging allow multiple processes to share a memory region?

c) Which physical address is referenced by the virtual address 0x00802BAD?

d) Which virtual address references the physical address 0x77777777?

e) Only the 20 most significant bits of a page directory entry are used to reference the location of a page table, the remaining 12 bits are used for flags. What does this imply for the location of page tables?

f) What does the kernel have to do so that different processes use different page tables?

g) If a memory reference takes 100 nanoseconds, how long does a paged memory reference take if there is no TLB or cache?

a) If a the page table of a process does not contain an entry that maps one or more virtual addresses to a physical address, the process can not access that physical address. Furthermore, the flags in the page directory and the page table entries can indicate that a page can only be accessed by ring 0 (kernel mode).

b) Both processes must have entries in their page tables which map a virtual address to the same physical address. Note that the virtual addresses can be different for both processes.

c) The address 0x00802BAD is split in three parts, the first 10 bits index a page directory entry, the next 10 bits index an entry in the page table which the page directory entry points to, and the last 12 bits are the offset into the page referenced by the page table entry. For our address 0x00802BAD the first 10 bits are 0000000010, so we refernce the second page directory entry, which points us to the page table at address 0x2BAD1000. The next 10 bits of the address are also 0000000010 which point is to the page starting at address 0x37777000. Now we add the last 12 bits of the address and get 0x37777BAD.

d) The physical address 0x77777777 is inside of the page starting at 0x77777000. This page has index 2 in the page table with the index 0. Therefore it is referenced by the virtual address 0x00002777.

e) Page tables are always placed at 4K-boundaries, because the last 12 bits of the address are zero. The same is true for pages itself, as page table entries also only use 20 bits for the address.

f) When a different process is scheduled, the contents of the page directory base register (PDBR) are changed, so that it points to the page directory of the scheduled process.

g) We have to read the page directory entry, the page table entry and read/write the data itself. So 3 times 100 ns = 300 ns.

1.3 Virtual Memory

Consider a paged virtual address space composed of 1024 pages of 4 KB each, which is mapped into a 1 MB physical memory space.

a) *What is the format of the logical address; i.e., which bits are the offset bits and which are the page number bits? Explain.*

Each page size is 4 KB = 2^{12} Bytes. Hence, we need 12 bits in order to represent the offset and to be able to access each byte in the page. Since we have 1024 pages we need 10 bits to represent all the pages numbers and to be able to access each page. Then, the format of the logical address is:

10 bits – page number	12 bits – offset

1.4 Page Replacement

Consider the following page access pattern:

```
1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6
```

How many page faults would occur for the following replacement algorithms, assuming one, two, three, four, five, six, or seven frames?

```
4 Frames (example):
LRU :
1 2 3 4 2 1 5 6 2 1 2 3 7 6 3 2 1 2 3 6
x x x x     x x       x x x     x
10 faults

FIFO :
1 2 3 4 2 1 5 6 2 1 2 3 7 6 3 2 1 2 3 6
x x x x     x x x x   x x x   x x   x
14 faults

Optimal :
1 2 3 4 2 1 5 6 2 1 2 3 7 6 3 2 1 2 3 6
x x x x     x x         x       x
8 faults
```

# Frames	LRU	FIFO	Optimal
1	20	20	20
2	18	18	15
3	15	16	11
4	10	14	8
5	8	10	7
6	7	10	7
7	7	7	7