UPPSALA UNIVERSITY
DEPARTMENT OF INFORMATION TECHNOLOGY

COMPUTER SYSTEMS/OPERATING SYSTEMS
Fall, 2007

IN-CLASS EXERCISE 7

Consider a simple paging system with the following parameters:

- 2^{32} byes of physical memory
- Page size of 2^{10} bytes
- 2^{16} pages of logical address space

1. How many bits are in a logical address?

2. How many bytes in a frame?

3. How many bits in the physical address specify the frame number?

4. How many entries in the page table?

5. How many bits in each page table entry? Assume each page table entry includes a valid/invalid bit.

In a virtual memory with paging system, the process' address space is divided into consecutive pages of fixed length: Page 0, Page 1, Page 2, etc. As such, a logical (or virtual) address consists of

- A page number
- An offset within that page

In a virtual memory with paging system, physical memory is divided into consecutive frames of fixed length (same length as the pages): Frame 0, Frame 1, Frame 2, etc. As such, a physical address consists of

- A frame number
- An offset within that frame

Some (but typically not all) of the logical address space, in the form of pages, is placed in frames in physical memory. For example, pages 3, 8, and 10 might be in frames 104, 8, and 378, respectively, with the other pages of the process on disk (or another form of secondary memory).

The goal of this exercise is to help you understand three methods for converting a logical address to a physical address (or a page fault if the page is not in memory):

- Paging
- Multilevel paging
- Paging with a translation lookaside buffer (TLB)

Real operating systems will often use a combination of these methods. In addition, there is interaction with the cache for main memory (see Figure 8.10 on page 345 of your text).

Throughout this exercise, logical addresses and physical addresses are written as pairs of numbers (*page number* | *offset* for logical addresses and *frame number* | *offset* for physical addresses). These are physically manifested as the high and low bits of a binary number.

Paging: Show the flow of information for paging and complete the examples below.

Address	Main memory	
	Present bit	Frame #
0	1	104
1	1	334
2	0	45
3	1	891
4	1	115
...	...	...
256	0	333
257	0	228
258	0	610
259	0	200
260	1	324
261	1	900
262	1	1005
263	0	820
264	1	20
265	0	5
266	0	1005
267	0	220
268	1	4
269	1	303
270	1	689
271	0	446
272	1	848
273	0	666
274	1	111
275	1	229

Suppose that the page table pointer is set to 256. Consider the following logical addresses. For each, determine whether or not a page fault is generated. If not, determine the physical address referenced.

2 | 105

5 | 640

12 | 320

Multilevel Paging: Show the flow of information for multilevel paging and complete the examples below.

Address	Main memory	
	Present bit	Frame #
0	1	104
1	1	334
2	0	45
3	1	256
4	1	115
5	1	713
6	1	512
7	0	228
...	...	...
256	1	900
257	1	1005
258	0	820
259	1	20
260	0	5
261	0	1005
261	0	220
...	...	...
512	1	303
513	1	689
514	0	446
515	1	848
516	0	666
517	1	111
518	1	229

Suppose that the root page table pointer is set to 0 and that the page number is divided as shown above.

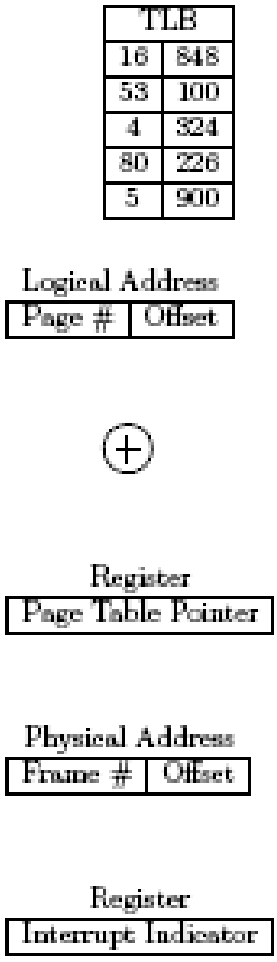
Consider the following logical addresses. For each, determine whether or not a page fault is generated. If not, determine the physical address referenced.

2 | 2 | 105

6 | 2 | 640

3 | 3 | 320

Paging with a TLB: Show the flow of information for paging with a TLB and complete the examples below.



Address	Main memory	
	Present bit	Frame #
0	1	104
1	1	334
2	0	45
3	1	891
4	1	115
...	...	...
256	0	333
257	0	228
258	0	610
259	0	200
260	1	324
261	1	900
262	1	1005
263	0	820
264	1	20
265	0	5
266	0	1005
267	0	220
268	1	4
269	1	303
270	1	689
271	0	446
272	1	848
273	0	666
274	1	111
275	1	229

Suppose that the page table pointer is set to 256. Consider the following logical addresses. For each, determine whether or not the page table entry is in the TLB and whether or not a page fault is generated. If a page fault is not generated determine the physical address referenced.

4 | 105

18 | 640

16 | 105