Operating Systems 2
Homework on Memory Management and Virtual Memory

Abhinav Gupta
ES15BTECH11002

April 4, 2018

This document is generated by LaTeX

1. (1 mark) Assuming a 2KB page size, find page number and offset for the following address reference: (3095)10

Sol.

The page number is 1 and the offset is 1047 assuming the pages are numbered from 0 onward.

Explanation: 110000010111 is the binary representation of 3095, the page size 2048 bytes, the last 11 bits of the above address determines offset and the first bit determines the page number.

2. (2 mark) Consider an OS with 32-bit virtual address and 24-bit physical address. System page size is 4KB. Find number of entries in an inverted page table.

Sol.
The number of entries in the inverted page table are 16777216 (2 power 24)

Explanation: There is a entry for each physical address in the inverted page table

3. (3 mark) An OS has following segment table.

Segment	Base	Length
0	229	420
1	85	105
2	5500	1400
3	1951	555
4	1000	321

For each logical address given below, find the physical address.
(a) 0, 345 (b) 3, 666 (c) 2, 876

Sol. and Explanation

a. Address valid(offset is less than length), $229 + 345 = 574$

b. Address not valid (offset more than length)

c. Address valid (offset less than length) $5500 + 876 = 6376$

4.(1 mark) What is double buffering? (max 2 sentence)

Sol.
If swapping onto a backing store is performed for process p1 and process p2 is brought in its place, where process p1 was having a pending I/O operation which got queued (maybe because of the disk being busy or the os serving other processes), this situation might lead the I/O operation to access the memory not of process p1.
To avoid this, one of the solution is of double buffering where I/O operations are executed only into operating system buffers.
Transfers between operation-system buffers and process memory then occur

only when the process is swapped in.

5. (2 marks) For the figure above, show the segment table.

Sol.

Segment	Base	Limit
0	1100	700
1	9350	550
2	5600	600
3	2200	2400
4	6200	2500

Explanation: Limit indicates the length of the segment, base is the starting physical address of the segment.

6. [1 mark] What is the use of page-length register. [max 2 sentence]

Sol.
Page length register is used to store the size of the page table.
This value is checked against every logical address to verify that the address is in the valid range for the process.

7. [2 mark] Show the address-subdivision for a 4-level hierarchical page table which has following properties: The outer-most page table has 536870912 entries. A second-level table has 8192 entries. A third level page table has 512 entries. A fourth-level table has 64 entries. The address is 64 bit.

Sol.
536870912 is 2 power 29, 8192 is 2 power 13, 512 is 2 power 9, 64 is 2 power 6, offset is therefore 7
Address subdivision is :

3rd outer page	2nd outer page	outer page	inner page	offset
p1	p2	p3	p4	d

p1 = 29
p2 = 13
p3 = 9
p4 = 6
d = 7

8. [4 mark] Given memory partitions of 310 KB, 580 KB, 380 KB, 220 KB, 790 KB, 600 KB and 155 KB (in order), how would the best-fit and worst-fit algorithms place processes of size 125 KB, 550 KB, 378 KB, 210 KB, 510KB and 400 KB (in order)?

Sol.
Best fit :
310 - x
580 - 550
380 - 378
220 - 210
790 - 400
600 - 510
155 - 125

Worst fit:
310 - x
580 - 378
380 - 210
220 - x
790 - 125
600 - 550
155 - x

Note: 510 and 400 memory blocks cannot be allocated with the worst fit algorithm

9.[2 mark] Consider a paging system, where the page table is also stored in the memory. (a) If a memory reference takes 150ns, how long does it take to complete a paged memory reference. (b) Assume we now use a TLB, which has a hit-latency of zero second (i.e., if the data is present in TLB, it is found instantaneously). Assume its hit ratio is 75%. Now, what is the effective memory reference time.

Sol.
a. It takes 2 memory accesses to fetch the data in case of paged memory reference, giving us a total of 300ns, first access memory for the page table and frame number (150ns) and then access the desired byte in memory(150ns).
b. 0.75 * 150 + 0.25 * 300 = 187.5 ns.
Note: TLB only requires the time to get the desired byte from memory when there is no time in searching the TLB.

10. [2 mark] Consider a hashed-page table. The average length of each hash chain is P.
On searching in this chain, an element is found (on average) on searching P/2 entries. The hash chain uses pointers to locate the next entry, as discussed in the class. Find the number of memory accesses required with the hashed-page table to serve a memory access completely.

Sol.
Note: It takes separate memory access for each entry in the linked list of a hash table entry as we need to fetch the memory address pointed by a entry in the linked list.(in case we don't find the desired virtual page number in the current entry)
It takes a memory access to get the hash table itself.
It takes in the worst case
time to access memory (to retrieve hash table) + P*time to access memory + time to access memory(to retrieve desired byte in memory) .
For the average case
It takes
time to access memory(to retrieve hash table) + (P/2)*time to access memory + time to access memory(to retrieve desired byte in memory).