CS342 Operating Systems – Fall 2018
Homework 2

Assigned: Dec 30, 2018
Due date: Jan 05, 2019, 23:55

Q1) Consider the following program pseudo-code. When we run this program, which distinct values will be printed out and how many times?

```
int  x = 1000;
int  y = 2000;
void *foo(void *p)
{         int y= (*((int*)p)) * 100;
          x = x + 100;
         y = y + 200;
          printf (“%d %d\n”, x, y);
          pthread_exit(NULL);
}

int main()
{
          pthread_t tids[3];
          int ret, i;

          printf (“%d %d\n”, x, y);

          ret = fork();
          if (ret == 0)
                    x = x + 5000;
          else
                    y =  y + 3000;

          printf (“%d %d\n”, x, y);
          for (i = 1; i < 2; ++i)
                    pthread_create (&tids[i], NULL, &foo, (void *) &i);
          for (i = 1; i < 2; ++i)
                    pthread_join(&tids[i]);
          printf (“%d %d\n”, x, y);
}
```

Q2) There are N processes in a ready queue in this order: N, N-1, …, 2, 1. That means Process N is at the head of the queue; Process 1 is at the tail. The CPU time required (CPU burst) by process k is k time units (1 <= k <= N).

i) [5 pts] Compute the average waiting time if FCFS scheduling is applied.

ii) [5 pts] Compute the average waiting time if SJF scheduling is applied.

Iii) [10 pts] Compute the average waiting time if RR(q=1) scheduling is applied.

Q3) Assume we have the following address division scheme in a system where virtual addresses are 36 bits and 3 level paging is used: 10 8 8 10. That means offset (displacement) is 10 bits. Answer the following questions: a) What is the page size? b) Assume a program has a virtual memory layout at time *t* as the following: It has 200 MB used at the beginning of the virtual memory and 140 MB used at the end of the virtual memory.

```
VM:       |200 MB | ………unused portion………| 140 MB |

          <---------------2^36 Bytes ------------>
```

How many second level page tables have to be used? How many third level page tables have to used?

Q4) Consider a computer that is uses segmentation and paging. The segment table of a process is the following (there are four segments):

Segment	Base	Length
0	1024	1024
1	4196	512
2	128	256
3	2048	768

Assume page size is 64 bytes. Assume virtual addresses are 16 bits long. Assume physical addresses are also 16 bits long. Assume a page i is located in a frame i+10 (for example, page #11 of linear logical memory is in frame #21 of physical memory). Convert the following logical addresses to their physical ones:

a) (0, 50)
b) (1,0)
c) (1,100)
d) (1,700)
e) (2,10)
f) (3,200)

All numbers above are decimal.

Q5) Assume memory access time is 200 ns. When a page fault occurs it takes on the average 10 ms to serve it. Assume we can tolerate at most 500 ns average memory access latency. What should be the page fault rate at most? Ignore the effect of dirty pages.

Q6) Consider an inode-based hypothetical file system (using indexed allocation) that is using bitvector for free block information. It sits in a 128 GB disk. The inode for a file contains 64 direct data block pointers, 1 single-indirect pointer, one double-indirect pointer, and one triple indirect pointer, similar to Linux/Unix file system. Assume inode is cached in memory (no disk access required to access inode), but no block is cached (no matter whether it is an index block or a data block). Index and data blocks are always accessed from disk no matter how frequently they are accessed. The block size is 512 bytes and pointer size is 8 bytes. Answer the following questions. Do not consider inode as an index block in this question.

a) Find out the maximum file size that can be supported.

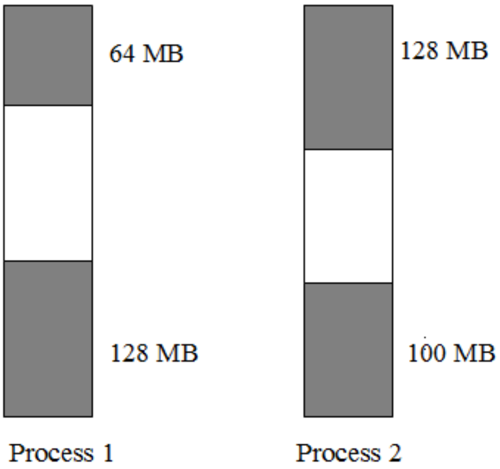
b) There are three files, A, B, and C. The size of file A is 64 KB, the size of file B is 192 KB, and the size of file C is 4 MB. Find out the number of index blocks required for file A, for file B, and for file C.

c) Find out is the average (i.e., expected) random access (uniform random) latency to file A, to file B, and to file C, assuming a disk access takes 10 ms time.

d) Assume the first 5 logical blocks of file A are sitting on disk blocks 20050, 20035, 21042, 35061, and 30253, respectively. Compute the physical location of offset (byte) 1700 of file A.

e) Find out the amount of space (in bytes, or KBs, or MBs) required in disk to keep the bitvector information.

Q7) Consider a computer that uses 36 bits long virtual addresses and 16 KB long pages. The computer has 4 GB RAM (main/physical memory). The physical addresses are also 36 bits long. A page table entry (no matter what kind of a page table it is in) is 8 bytes long. We have 2 processes in the system with the following logical memory layouts (dark areas are used):



a) What is the memory (RAM) space required to hold *all* page table information for these two processes if single level page table scheme is used.

b) What is the memory space required to hold *all* page table information for these two processes if two level page table scheme is used with the following address partitioning: 11 bits for p1, 11 bits for p2, 14 bits for offset (i.e., 11,11,14)?

c) What is the memory space required to hold *all* page table information for these two processes if inverted page table scheme is used (ignore chaining effects)?