CSc 360: Operating Systems (Summer 2012)

Written Assignment

Due: 23:55 pm, Aug. 5, 2012

Remarks

This assignment is on paper and includes four parts. It serves two purposes: (1) encourage you to explore what has been omitted in class due to time constraint, (2) help you in the preparation of the final exam.

You need to submit your solution in a pdf file to connex dropbox by the due time. No extension will be granted.

1 Part 1: OS overview and system calls

1. (5%) Some OS like Palm OS provides no means of concurrent processing. Briefly discuss three major complications (in a bullet format) that concurrent processing adds to an operating system.

2. (10%) In the following example, assume all system and library calls complete without error.

```
#define OUTPUT printf(``%d '', i)

main() {
  int i=10; OUTPUT;

  if (fork()) {
    i+=5; OUTPUT;
  } else {
    i+=3; OUTPUT; return(0);
  }
}
```

   (a) Please write down all possible outputs when running this program.

   (b) Add one system call in the pseudo code to ensure that the output values are always in increasing order.

2 Part II: IPC, Process Synchronization and CPU scheduling

3. (5%) How does the signal() operation associated with monitors differ from the corresponding operation defined for semaphores?

4. (15%) Consider a system consisting of processes $P_1, P_2, \ldots, P_n$, each of which has a unique priority number. Write the pseudo-code of a monitor that allocates three identical line printers to these processes, using the priority numbers for deciding the order of allocation.

5. (15%) A clinic consists of a waiting room with n chairs and a doctor. If there is no patient to be served, the doctor becomes idle. If a patient comes to the clinic and all chairs are occupied, the patient leaves the clinic. If the doctor is busy but chairs are available, the patient sits in one of the free chairs. Of course, we require that the doctor cannot be idle if there are waiting patients. Write the pseudo-code to coordinate the doctor and the patients.

3 Part III: File Systems

6. (5%) Assume we are using the Rhinopias disk drive discussed in class (refer to the lecture slides for specification).

   (a) What is the maximum transfer rate for a single track?

   (b) What is the maximum transfer rate for accessing five tracks consecutively (assume head skewing)?

7. (5%) Would log-structured file systems have been feasible in the early 1980s when FFS was designed? Explain. (Hint: Consider the motivation of log-structured F.S.)

8. (5%) Explain how renaming a file can be done using the consistency-preserving approach so that a crash will not result in the loss of the file. Be sure to take into account the link count stored in the file's inode. Note that there may have to be a temporary inconsistency; if so, explain why it will not result in lost data.

9. (5%) What are your recommendations if you were required to support long file names in FAT12?

4 Part IV: Main Memory and Virtual Memory

10. (5%) Explain the difference between internal and external fragmentation.

11. (5%) Given five memory partitions of 100 KB, 500 KB, 200 KB, 300 KB, and 600 KB (in order), how would each of the first-fit, best-fit, and worst-fit algorithms place processes of 212 KB, 417 KB, 112 KB, and 426 KB (in order)? Which algorithm makes the most efficient use of memory?

12. (5%) Assuming a 1 KB page size, what are the page numbers and offsets for the following address references (provided as decimal numbers):

(a) 2375

(b) 19366

(c) 30000

(d) 256

(e) 16385

13. (5%) Explain why we need Inverted Page Tables and how they work?

14. (10%) A computer whose processes have 1024 pages in their address spaces keeps its page tables in memory. The time required for one word reading via the page table is 10 nsec. To reduce the access time, the computer uses a TLB. The time required for one word reading via the TLB is 3 nsec, which includes 1 nsec for TLB lookup. If we assume there are no page faults, what TLB hit rate is needed to reduce the mean one word reading time to 5 nsec?