Course ECE354

Final Exam

Aug. 11, 2009

1. Two parents and their two children are standing in front of a cookie dispenser. The parents feed the machine with one cent coins. The machine uses the coins to make cookies, and dispenses them once the deposited amount exceeds three cents. The children continuously stares at the cookie machine and whenever it sees a cookie, then they will grab and eat it.

   Available functions include: Parents use *PrepareCoin()* to prepare the next coin to be inserted and *InsertCoin()* to deposit a one cent coin in the machine. The machine uses *PrepareCookie()* to bake the next cookie and *DispenseCookie()* to dispense the baked cookie. The kids use *GrabCookie()* to take the dispensed cookie and *EatCookie()* to eat the taken cookie. Assume that the usual functions mentioned in the book such as *parbegin(...)* are available. Assume that *PrepareCoin()* takes a random amount of time, often much longer than all the other functions. The cookie machine accepts new coins as it produces the cookie (i.e., parent can execute *InsertCoin()* while the machine executes *DispenseCookie()*).

   Make sure that (a) the kids take/eat only when a cookie is available, (b) the machine only dispenses cookies when enough funds have been deposited, (c) the parents never try to insert coins simultaneously, (d) the kids don't touch the dispenser unless the machine finished dispensing the cookie, (e) only one kid at a time grabs a cookie. The world is an unfair place, so don't bother with fairness when giving cookies to the kids.

   Write the pseudo-code program that controls concurrency for above mentioned scenario and explain the program. Don't use busy waiting. *[25p]*

2. The computer has three memory frames. The running program requests pages in the following order: 2,3,2,4,1,5,2,4,5,3,2,5, and 2. Show the allocation of pages to frames after each page using the clock policy with one use bit and compare it to using the clock policy with zero use bits. Justify why pages get removed from memory. *[15p]*

3. Scheduling:

   (a) Describe one non-preemptive and one preemptive (any form of preemptive) scheduling policy and discuss their effect on processes and whether starvation is possible. *[6p]*

   (b) Create a process scheduling example with at least five processes. Specify the arrival time and service time for each process and create the schedule for the

two policies you listed above. Have at least two preemptions appear in the schedule. *[6p]*

4. Name and explain three file types usually found in a UNIX-like file system? *[6p]*
5. Explain the term *reentrant procedure* and provide an example of a procedure that is non-reentrant. *[10p]*
6. Explain the difference between the producer/consumer and the reader/writer problem in concurrency control and how it affects the solution to each problem. Specifically mention: explain and and given an example of each, the distinguishing characteristics, and how their concurrency solutions differ. No need to provide a code example for each, but if it helps your argument, feel free to write it down. *[15p]*
7. List and explain an assembly instruction that will take the processor from the user mode into the supervisor mode. *[3p]*
8. Explain for the file ogranization termed *pile:*
   (a) Its data organization *[6p]*
   (b) Two advantages when using it *[4p]*
   (c) Two disadvantages when using it *[4p]*
9. Priority inversion and one solution:
   (a) Explain the term priority inversion *[3p]*
   (b) Provide an example that demonstrates this effect *[6p]*
   (c) Show how priority inheritance solves the problem in your example *[6p]*
10. Microkernel:
    (a) Explain the architectural style of a microkernel-based operating system *[4p]*
    (b) Explain three advantages of this style *[6p]*
11. Explain how system calls work and refer to your lab project. Mention where and how you pass the boundary from user space into kernel space, how arguments and return values are passed to system calls, and why context switches among threads are faster in user-level threads than in kernel-level threads. *[12p]*

End of exam. Total points: 137