Course ECE354

Replacement Final Exam

Dec. 2009

1. Two parents and their three 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 two 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 using semaphores for above mentioned scenario and explain the program. Don't use busy waiting. *[25p]*

2. The computer has four memory frames. Define a sequence of ten memory accesses. Show the allocation of pages to frames after each page using the LRU policy and compare it to the optimal policy. The sequence should cause at least four page faults in LRU and less than that in the optimal one. Explain why pages get removed from memory. *[15p]*

3. Explain plain round robin scheduling and under what circumstances it performs poorly. Propose a fixe for one of its problems. *[12p]*

4. How and why can one convert the clock policy for memory allocation into a FIFO policy? *[5p]*

5. Explain the term *reentrant procedure* and provide an example of a procedure that is non-reentrant. *[10p]*

6. Explain the term *deadlock* and provide a small code example that leads to a deadlock. *[10p]*

7. Why is it that even if you store only one bit in a file, the file will still measure several kilobytes in size? Explain in a couple of sentences. *[7p]*

8. Explain the TRAP call on the lab hardware and how it is used for system calls. *[5p]*

9. Draw the seven state diagram for process state transitions. Explain the nature of all transitions that leave the Ready state in one sentence each. *[11p]*

End of exam. Total points: 100