A Brief Review for the Final

These slides are created by Dr. Huang of George Mason University. Students registered in Dr. Huang’s courses at GMU can make a single machine readable copy and print a single copy of each slide for their own reference as long as the slide contains the copyright statement, and the GMU facilities are not used to produce the paper copies. Permission for any other use, either in machine-readable or printed form, must be obtained form the author in writing.

Final Examination

- 4:30pm July 28th
- Location: David King Hall Room 1006
- Accumulative; approximately 70% after midterm
- Some problems may be derived from projects, but you are not required to memorize details.
- Close book. You can use two letter-size cheating sheets written by hand. Machine printouts will be confiscated.
Learn all course materials by heart; use cheating sheets as a last resort.

You can use a calculator, pens, erasers. No other equipments allowed.
  – Do not bring cell phones or beepers.
  – You lose 10 points each time your cell phone rings.

All special requests about the final exam and course grade must be made before the final.
  – No exceptions. Period.

Last Office Hours

There will be a last office hour after the final exam.

Time and date to be announced on the course web page before final.

Before the last office hour you will receive an email giving
  – Your record of the entire semester
  – The formula to calculate course grades
  – Your course grade
- In the last office hour, you can
  - Pick up the final exam (and argue with grading)
  - Fix any problems you may have before I submit course grades
- After that, I change grades only for outright mistakes on my part.

Reader Writer Problem

```
wait(wrt);
...
writing
...
signal(wrt);
```

```
wait(mutex);
readcount++;
if (readcount == 1) wait(wrt);
signal(mutex);
...
reading
...
wait(mutex);
readcount--;
if (readcount == 0) signal(wrt);
signal(mutex):
```
Given the Reader-Writer algorithms on the next page,
- How is one-writer rule enforced?
- What problem would occur if the mutex semaphore is removed?

Rewrite the project 2 with critical region.
- Share data
  
  Struct shared_area {
    char cmd; int n; int result;
  }

  - Give the critical region code for the foreground process
  - Give the critical region code for the background process.
We protect a critical section by a semaphore as follows:
- Wait (mutex)
- Critical Section
- Signal (mutex)

Show that if the wait and signal are not performed atomically, more than one process may enter the critical section.

Semaphore Implementations

wait(S):
    S.value--; 
    if (S.value < 0) {
        add this process to S.L; 
        block; 
    }

signal(S):
    S.value++; 
    if (S.value <= 0) {
        remove a process P from S.L; 
        wakeup(P); 
    }
Argue that the algorithm below satisfies the three requirements of critical sections: 1) mutual exclusion, 2) progress, 3) bounded delay

```c
    do {
        flag [i]:= true;
        turn = j;
        while (flag [j] and turn = j) ;
            critical section
        flag [i] = false;
            remainder section
    } while (1);
```

Explain in what way IDE drives causes more CPU workload than their SCSI counterpart

Circle the item(s) that are represented as a file on Unix platforms
- Mouse
- CD burner
- Semaphores
- Printer
- Shared memory segments
- An entire hard drive
- Explain the concept of journaling file systems. Your answer must cover
  - Transaction
  - Commit
  - Crash recovery
  - Benefits

- Explain how the Reiser file system avoids writing the contents of large files twice. Since the data are not written in the journal, will the file system be left in inconsistent state if the system crashes in the middle of writing such files?

- What is an I-node?

- Explain the concept of transactions.

- A user is creating a new, empty file on the Reiser file system. Give the disk operations that would form the transaction.
- Explain direct IO and memory-mapped IO.
- Explain programmed IO and discuss its advantages and disadvantages.
- Explain polling and give its disadvantages.
- Explain the concept of RAID level 5.
  - Compared to level 4, what problem does it solve?
  - Discuss its advantages in serving concurrent small read requests
  - Discuss its disadvantages in serving small write requests.

Extra Credit Project

- Modify Project #3 so that it allows for multiple concurrent “viewer” processes.
- Foreground and background processes perform the same functions as in Project #3.
- A new viewer.c program implements executable viewer, launched as follows
  - viewer shm_key, sem_key1, sem_key2 …
- A viewer process prints the values of n, cmd and results on screen every second.
  - Use the sleep() system call.
Both the foreg and backg are considered writers to (n, cmd, result).

Viewers are readers of (n, cmd, result).

When the foreg write a new command, it must set result to -1.

You must make sure that only one writer can be in critical sections at a time.

When there is no writer, more than one view can be in critical sections.

When testing, launch at least 3 viewers.
  – They all quit when seeing the ‘q’ command.

You can use more shared variables and/or semaphores.
Submission

- Due midnight July 30th
- Send emails to the TA with the following attachments:
  - foreg.cc
  - backg.cc
  - viewer.cc
  - Makefile
  - readme

- When the TA issues “make all”, three executables are produced: foreg, backg, and viewer.
- Describe in the readme:
  - How to launch foreg, backg, and viewer.
  - The foreg must print out sufficient info to launch backg/viewer processes.