School of Computer Science
Faculty of Science
COMP-2560: System Programming, Fall 2022

Lec#	Date	Title	Due Date	Grade Release Date
Lec06	Week 06	**File System**	Oct. 26, 2022, Wednesday Midnight EDT	Oct. 31, 2022

The objectives of the weekly lecture assignments (Lecs) are to practice on topics covered in the lectures as well as improving the student's *critical thinking and problem-solving skills in ad hoc topics that are closely related but not covered in the lectures*. Lecture assignments also help students with research skills, including accessing, retrieving, and evaluating information (information literacy).

Lecture Assignments Deliverables

You should answer **two questions** below using an editor like MS Word, Notepad, and the likes or pen in papers. In the latter case, you must scan the papers clearly and merge them into a **single file** `lec06_uwindid.pdf` containing your name, uwinid, student#. **Please note that if your answers cannot be read, you will lose marks.** Please follow the naming convention as you lose marks otherwise. Instead of uwindid, use your own account name, e.g., mine is hfani@uwindsor.ca, so my submission would be: `lec06_hfani.pdf`

Lecture Assignments

Select two questions based on your preference!

1. Devices are categorized into two main groups: character devices and block devices. Explain the differences and the effects on the way we should read or write to them.
2. A modular design for a kernel allows it to update its component without broadcasting the change to other parts. For instance, we may have different File Systems in the market, and at any time, we can unplug the current File System and plug in a new one. Specifically, there are many Disk File Systems (part of the File System that handles storing files in storage devices) as the storage devices' technology has evolved dramatically in history. Pick two most recent Disk File Systems and compare them.
3. An image file that is stored by Unix File System (UFS) cannot be read by MS-Windows's NT-File System. Why? Is it possible to write a C program to recover the image? How?
4. Who connects the standard file descriptors 0, 1, 2 to normal files or devices for a program: shell or kernel? Can we change the assignment inside our program? Explain your answer.
5. The Bourne shell, Bourne-again shell, and Korn shell notation have a notation `digit1>&digit2`, which basically redirect file descriptor `digit1` to the same file as file descriptor `digit2`. What is the difference between the two commands shown below? (Hint: The shells process their command lines from left to the right.)
   a) `./main > outfile 2>&1`
   b) `./main 2>&1 > outfile`
6. Why would a program want multiple file descriptors to the same file or device using `dup()` or `dup2()` system calls?

7. When using `read()` or `write()` system calls, we need to specify a buffer. Should we select large buffer or small? What is the optimum size of buffer based on the underlying task?

8. Assuming two programs A and B are running in parallel on each processor of a computer system that has two processors. What is the problem if the kernel's File System successfully opens the same file for A and B for:
   a. reading only?
   b. reading for A and writing for B?
   c. writing only?

Do you think, kernel's File System should care and prevent/fix the problem? Justify your answer.