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

Lec#	Date	Title	Due Date	Grade Release Date
Lec11	Week 11	**IPC & Network IPC**	Nov. 30, 2022, Wednesday Midnight	Dec. 5, 2022

The objectives of the weekly lecture assignments (Lecs) are to practice on topics covered in the lectures as well as improve 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** lec11_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: lec11_hfani.pdf

Lecture Assignments

Select two questions based on your preference!

1. Is it possible for a process not to be stopped or killed by CTRL+Z or CTRL+C, respectively? If yes, how?
2. **[Serialization-Deserialization]** We can send or receive basic data types (char, int, long, float, …) in IPC and Network IPC because the system calls to send or receive information accepts void*. In the kernel, the information is transferred in a stream of bytes. When sending information, the kernel does serialize the information to an array of bytes. When receiving, the kernel does deserialize the array of bytes into the desired data type. For example, if we send an integer number 206, it becomes 4 bytes of [00000000 00000000 00000000 11001101]. When receiving the number, the 4 bytes are deserialized back into 206. In C program, we can add new data types using struct. For example, we can define a data type point that has two integer numbers x and y as the coordination in 2D space.

```
struct point{
    int x;
    int y;
}
struct point a;
a.x = 0;
a.y = 1;
```

   a) Is it possible to transfer the variable a of type point from the child to the parent (or vice versa) in IPC? Justify your answer.
   b) Is it possible to transfer the variable a of type point from a sender process to a receiver process (or vice versa) in Network IPC? Justify your answer.

3. [**Byte Order (Endianness[1])**] The byte order is a characteristic of the processor architecture, dictating how bytes are ordered within larger data types, such as integers, in memory. There are two well-known byte ordering:
   i) *Big-endian*; the least significant byte contains the lowest byte address. This is intuitive. For example, for the integer number 206, the 4 bytes [00000000 00000000 00000000 11001101] are stored in memory like:

0x3: [11001101]

[1] https://en.wikipedia.org/wiki/Endianness

0x2: [00000000]
0x1: [00000000]
0x0: [00000000]

ii) *Little-endian*; this is the opposite of little-endian. The highest byte address occurs in the least significant byte. For example, for the integer number 206:

0x3: [00000000]
0x2: [00000000]
0x1: [00000000]
0x0: [11001101]

a) When communicating with processes running on the same computer as in IPC, is byte ordering a problem? Justify your answer.
b) When communicating with processes running on different computers as in Network IPC, is byte ordering a problem? Justify your answer.

4. Is it possible for the receiver to send an acknowledgment to the sender about receiving a message in a connectionless communication (e.g., in UDP of TCP/IP)? If yes, how?
5. On the one hand, it is said that binding address for the sender/client process is optional. On the other hand, it is said that any communication in a computer network requires the addresses on both sides of the communication. Are these two sentences contradictory? Explain your answer.
6. [**Important**] Our Network IPC examples in class were run in the same machine because we could not have multiple computers available for our practice. That's why we put the same IP address for both sender/client and receiver/server processes. This implies that Network IPC can also be an alternative to IPC. In other words, in the same computer, instead of creating parent and child process and communication using pipe/FIFO/signal, we can create two separate processes using the same IP address but different ports and use Network IPC like UDP or TCP for communication. Compare the pros and cons of using Network IPC vs. IPC in *the same computer*.
7. In which of the following scenarios are sockets used?
   a) Connecting to the School's VPN
   b) Connecting to the School's server machine
   c) Opening Google's page from a browser
   d) Opening Facebook's app from a cellphone
   e) Sending an email to hfani@uwindsor.ca
   f) Receiving an email from hfani@uwindsor.ca
   g) Attending the virtual class in Blackboard
   h) Remote Desktop
   i) Copy-Paste of a text from MS-Word to MS-Exel