University of the Fraser Valley

COMP 340 - FINAL EXAMINATION
April 22nd, 2023

NAME: Zhiwei Qiu
FAMILY NAME GIVEN NAMES

UFV-ID #: 300191315

INSTRUCTIONS

1. This is a 24-hour take-home, **open-book exam**.
2. You can use the Internet, but you must **cite your references** and **you cannot copy-and-paste**.
3. **No collaboration is allowed**.
4. Place your **NAME and UFVID** above.
5. There are **100 points** in total.
6. There are **2 pages** including this cover sheet.
7. Your exam shall begin at the scheduled time of **2PM, Saturday, April 22nd, 2023, Vancouver time** (just begin when you receive it if a bit earlier).
8. When you are finished, upload your completed exam to Blackboard including screenshots and proof of compilation. **Include real C object code files** and do not cut-and-paste code into text files or pdfs.
9. Exams will not be accepted after 2PM, Sunday, April 23rd, 2023, Vancouver time.

1.

[10] Give an example to demonstrate that the four necessary conditions for a resource deadlock to occur are not sufficient conditions. When are these conditions sufficient for a resource deadlock to occur?

The four necessary conditions for a resource deadlock to occur are:
1. Mutually exclusive resources:
Only one process at a time can access mutually exclusive resources.These resources cannot be shared among multiple processes simultaneously.
2. Hold and wait:
A process may hold some resources and wait for additional resources at the same time. In this scenario, the process will not release the resources it is already holding. It will keep waiting while holding the resources it has already acquired. This can potentially lead to a deadlock.
3. Non-priority, non-preemptive resources:
Resources that have been acquired by a process cannot be taken away from that process by force. The process has to voluntarily release the resources when it is done with them.
4. Circular wait:
A group of two or more processes are stuck in a circular chain of waiting for each other to release resources. Each process in the group is waiting for another process in the same group to release a resource. This results in a deadlocked state where none of the processes can progress forward.

These conditions are necessary but not sufficient for a deadlock to occur. For example, consider a system with 3 processes P1, P2 and P3 and 3 resources R1, R2 and R3. The allocation of resources is:
P1 - R1
P2 - R2
P3 - R3
Even though the 4 conditions hold, there is no deadlock as there is no circular wait. Each process is holding one resource and waiting for other resources held by other processes, but the wait graph is acyclic.
For a deadlock to actually occur, the necessary conditions must lead to a circular wait where each process is waiting for a resource held by the next process in the set. Only when these conditions manifest as a circular wait graph, they become sufficient for a deadlock to occur.
In short, necessary conditions point to situations where a deadlock is possible, but for an actual deadlock to occur, the wait graph must be circular - this transforms the necessary conditions into sufficient ones for deadlock occurrence.

2. **[25]** Construct the Banker's algorithm for multiple resources that compiles in C. It must be based only on the description given in your textbook. Any copying and pasting of code will result in an automatic grade of zero on this exam. Be sure to comment thoroughly throughout your code describing each major step in detail. (Attach your source code and screen shots to demonstrate that it compiles).

[10] In a virtualization environment, the I/O operations of I/O devices of a guest OS are translated by the hypervisor. This behavior results in a negative performance impact. Discuss how this problem is remedied in detail.

3.

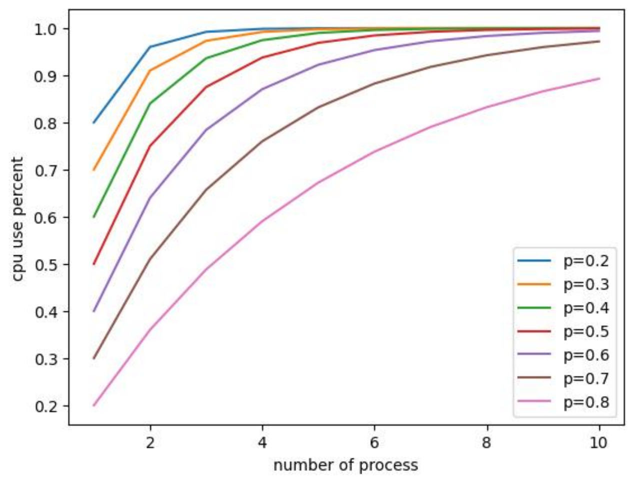
For the first remedial measure, namely I/O virtualization, the system management program forges I/O devices, so that the customer operating system does not have to directly access the actual physical I/O devices. The system management program intercepts I/O requests from the client operating system and processes them internally, then returns the response to the client operating system. This reduces the dependency of the guest operating system on physical I/O devices, but does incur some performance overhead.
The second remedial measure is called PCI passthrough. This will directly allocate physical I/O devices to the guest operating system, completely bypassing the system management program. This allows guest operating systems to directly access I/O devices, avoiding the performance impact of system management programs converting I/O requests. PCI passthrough is typically used for performance sensitive I/O devices such as GPUs, network interface cards, etc.

[25] A company, using a multi-programming operating system, has 1 megabyte of memory. The operating system occupies 250K of memory and every process that is executed also requires 250K of memory. The processes have an average I/O wait time of 80%. The company asks you if they should invest in more memory and, if so, how much. Show your answers graphically. Show your calculations behind your suggestions and do not just say as much RAM as possible.

a) What would you advise and why?
b) Would your advice change if the company said they had made a mistake and the average I/O wait time was only 20%? If so, why?

6.

Assume that that a process spends *p* percent of its time waiting for I/O. With *n* processes in memory the probability that all *n* processes are waiting for I/O (meaning the CPU is idle) is p^n.
The CPU utilisation is then given by
CPU Utlisation = $1 - p^n$
So we can plot the relationship between CPU utilization and the number of processes under different I/O ratios, as shown in the following figure



A)

Initially, we ran three processes. If we add another megabyte, the CPU utilization will increase from 49% to 70%. I think it's worth it. Adding another megabyte will result in a 91% utilization rate. I also think it's worth it.
An additional megabyte (allowing 15 processes to run) will increase our utilization rate to 96%.
I suggest that the company increase its memory by two megabytes (with a utilization rate of 91%).

B)

If we consider the case where the I/O waiting time is 20%, we can obtain the following numbers.
IO waiting time=20%
Number of processes CPU utilization

3	0.99
7	0.99
11	0.99
15	1.0

Even with three processes running, the CPU utilization rate reaches 99%, and I don't think it's worth adding more memory.