UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Faculty of Engineering, Built Environment and Information Technology

Fakulteit Ingenieurswese, Bou-omgewing en Inligtingtegnologie / Lefapha la Boetšenere, Tikologo ya Kago le Theknolotši ya Tshedimošo

Make today matter

www.up.ac.za

Department of Computer Science

COS 122 Operating Systems

Assignment 3

Due: 06 October 2022 @ 23:00

PLAGIARISM POLICY

UNIVERSITY OF PRETORIA

The Department of Computer Science considers plagiarism as a serious offence. Disciplinary action will be taken against students who commit plagiarism. Plagiarism includes copying someone else's work without consent, copying a friend's work (even with consent) and copying material (such as text or program code) from the Internet. Copying will not be tolerated in this course. For a formal definition of plagiarism, the student is referred to `http://www.ais.up.ac.za/plagiarism/index.htm` (from the main page of the University of Pretoria site, follow the *Library* quick link, and then click the *Plagiarism* link). If you have any form of question regarding this, please ask one of the lecturers, to avoid any misunderstanding. Also note that the OOP principle of code re-use does not mean that you should copy and adapt code to suit your solution.

Objectives

This assignment evaluates the understanding and application of various key concepts and functions found in computer and operating systems. It covers chapters 5 and 6 of the prescribed textbook. This assignment has 4 tasks for a total of 20 marks.

Upload Instructions

You need to provide written answers to the tasks in this assignment. You are then required to submit a document containing these answers in order for them to be marked. Show all the intermediate and calculation steps in your answers (excluding the multiple choice task). Some marks will be awarded for intermediate steps.

- Upload your document to the Assignment 3 assignment slot on COS 122 ClickUP before 23:00 on 06 Oct-2022. **No late submissions will be accepted!**
- All documents must be in either text, Word or PDF format (typed not handwritten) as **no other formats** will be marked.
- **Failure to upload your answers will result in 0 marks being awarded for your assignment!**

Task 1 .... (5 marks)

1.1 Which situation occurs when a runnable process is overlooked indefinitely by the scheduler, although it is able to proceed? (1)

A. Deadlock
B. Livelock
C. Starvation
D. Mutual Exclusion

1.2 Which situation occurs when multiple threads or processes read and write a shared data item, and the final result depends on the relative timing of their execution? (1)

A. Race Condition
B. Livelock
C. Atomic Operation
D. Mutual Exclusion

1.3 Which strategy of deadlock is to design a system that dynamically determines if resource allocation will produce deadlock? (1)

A. Deadlock Detection
B. Deadlock Prevention
C. Deadlock Avoidance
D. Deadlock Diversion

1.4 What deadlock condition allows only one process to use a resource at a time, and no process may access a resource unit that has been allocated to another process? (1)

A. Circular Wait
B. No Pre-emption
C. Mutual Exclusion
D. Hold and Wait

1.5 Which mechanism allows multiple threads to have simultaneous read-only access to an object protected by the mechanism? (1)

A. Semaphore
B. Mutex
C. Condition Variable
D. Readers/Writer Lock

Task 2 .... (4 marks)

Consider the following implementation of the producer/consumer problem using counting semaphores:

```
/* program producer/consumer */
semaphore items = 0, free = 1;
void producer()
{
    while (true) {
        produce();
        semWait(free);
        append();
        semSignal(items);
        semSignal(free);
    }
}
void consumer()
{
    while (true) {
        semWait(free);
        semWait(items);
        take();
        semSignal(free);
        consume();
    }
}
void main()
{
    parbegin (producer, consumer);
}
```

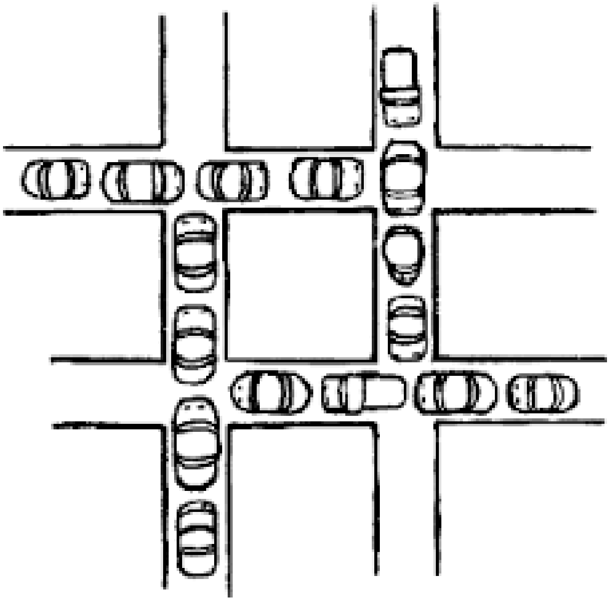
2.1 What problem can occur with this implementation? (1)

2.2 Give an execution sequence that shows the problem and provide an explanation. (2)

2.3 What would you change in the code to fix the identified problem? (1)

Task 3 .... (5 marks)

Given the intersection scenario in the figure below, answer the following questions:

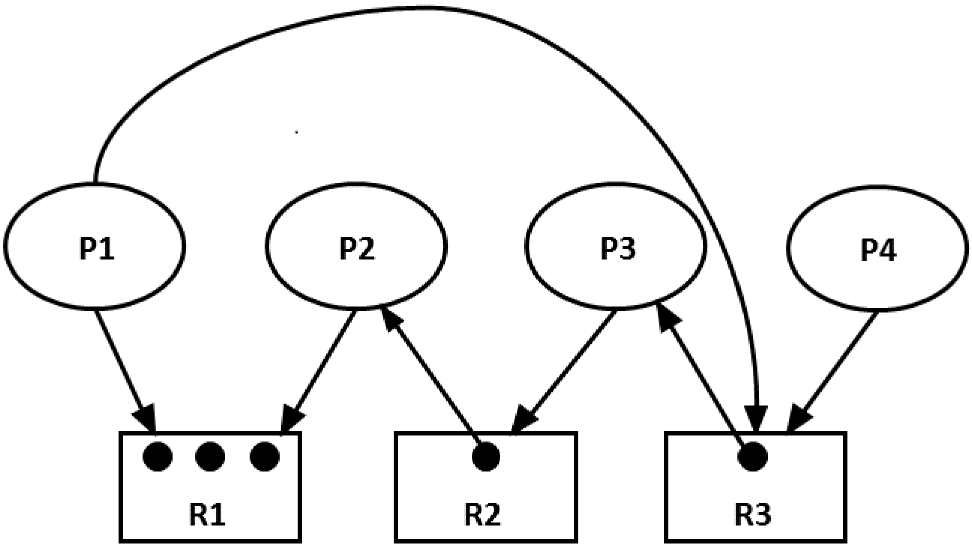


3.1 Explain, based on each of the four deadlock conditions, why there might be a deadlock situation. (4)

3.2 Name a synchronisation mechanism that could be implemented at the intersection in order to make a deadlock impossible. (1)

Task 4 .... (6 marks)

Given the resource allocation graph below, answer the following questions:



4.1 Identify if the processes can reach a deadlock state. If yes, which processes will be deadlocked in this state? Justify your answer. If no, why is it impossible to reach a deadlock? (2)

4.2 Provide the total claim matrix C, the resource vector R, the available vector V and the allocation matrix A for the state depicted in the given resource allocation graph. (4)