**FALL 2017 COMP 3511 Homework Assignment**

**Handout Date: October 30, 2017 Due Date: November 11, 2017**

Name: SHI Zhenmei ID: 20329335 E-Mail: zshiad@connect.ust.hk

**Please read the following instructions carefully before answering the questions:**

* You should finish the homework assignment **individually**.
* There are a total of **3** questions.
* When you write your answers, please try to be precise and concise.
* Fill in your name, student ID, email and Section number at the top of each page.
* Please fill in your answers in the space provided, or you can type your answers in the MS Word file.
* **Homework Collection: the hardcopy** is required and the homework is collected in collection box #20. The collection boxes locate outside Room 4210, near lift 21 (there are labels attached on the boxes).

1. (20 points) Multiple choices
   1. All processes share a semaphore variable mutex, initialized to 1. Each process must execute wait(mutex) before entering the critical section and signal(mutex) afterward.

C

i) Suppose a process executes in the following manner :

signal(mutex);

…..

critical section

…..

wait(mutex);

In this situation :

A) a deadlock will occur

B) processes will starve to enter critical section

C) several processes maybe executing in their critical section

D) All of these

B

* 1. A deadlock free solution to the dining philosophers problem :

A) necessarily eliminates the possibility of starvation

B) does not necessarily eliminate the possibility of starvation

C) eliminates any possibility of any kind of problem further

D) None of these

A

* 1. A system is in the safe state if

A) the system can allocate resources to each process in some order and still avoid a deadlock

B) there exist a safe sequence

C) both (a) and (b)

D) none of the mentioned

D

* 1. To ensure no preemption, if a process is holding some resources and requests another resource that cannot be immediately allocated to it :

A) then the process waits for the resources be allocated to it

B) the process keeps sending requests until the resource is allocated to it

C) the process resumes execution without the resource being allocated to it

D) then all resources currently being held are preempted

A

* 1. One way to ensure that the circular wait condition never holds is to \_\_\_\_\_\_\_\_.

A) impose a total ordering of all resource types and to determine whether one precedes another in the ordering

B) to let a process wait for only one resource at a time

C) to never let a process acquire resources that are held by other processes

D) All of these.

B

* 1. A system with 5 processes P0 through P4 and three resource types A, B, C has A with 10 instances, B with 5 instances, and C with 7 instances. At time t0, the following snapshot has been taken :

Process

P0

P1

P2

P3

P4

Allocation (process-wise : P0 through P4 top to bottom)

A B C

0 1 0

2 0 0

3 0 2

2 1 1

0 0 2

Max (process-wise : P0 through P4 top to bottom)

A B C

7 5 3 7 4 3 ->7 5 5

3 2 2 1 2 2-> 5 3 2

9 0 2 6 0 0

2 2 2 0 1 1-> 7 4 5

4 3 3 4 3 1-> 5 3 4

Available

A B C

3 3 2

The sequence leads the system to :

A) an unsafe state

B) a safe state

C) a protected state

D) a deadlock

B

* 1. If we preempt a resource from a process, the process cannot continue with its normal execution and it must be :

A) aborted

B) rolled back

D) terminated

D) queued

A

* 1. Run time mapping from virtual to physical address is done by

A) memory management unit

B) CPU

C) PCI

D) none of the mentioned

C

* 1. If there are 32 segments, each of size 1Kb, then the logical address should have :

A) 13 bits

B) 14 bits

C) 15 bits

D) 16 bits

C

* 1. Consider a computer with 8 Mbytes of main memory and a 128 K cache. The cache block size is 4 K. It uses a direct mapping scheme for cache management. How many different main memory blocks can map onto a given physical cache block ?

A) 2048

B) 256

C) 64

D) 8

1. (30 points) Please answer the following questions in a few sentences
   1. (6 points) List the factors on the basis of which the process to be aborted is chosen.
      * 1. Priority of the process
        2. How long process has computed, and how much longer to completion
        3. Resources the process has used
        4. Resources process needs to complete
        5. How many processes will need to be terminated
        6. Is process interactive or batch?
   2. (5 points) Discuss the tradeoff in terms of fairness and throughput in the solution to the first readers-writers problem. Propose a method for solving the readers- writers problem without causing starvation.
      * 1. Throughput in the readers-writers problem is increased by favoring multiple readers as opposed to allowing a single writer to exclusively access the shared data.
        2. However, favoring readers could result in starvation for writers, which leads to unfairness.
        3. The starvation in the readers/writers problem could be avoided by keeping timestamps associated with waiting processes. When a writer is finished with its task, it would wake up the process that has been waiting for the longest duration. When a reader arrives and notices that another reader is accessing the database, then it would enter the critical section only if there are no waiting writers. These restrictions would guarantee fairness.
   3. (4 points) Compare the circular-wait scheme used in deadlock prevention with the deadlock-avoidance schemes (like the banker’s algorithm) with respect to the following two issues:

a. Runtime overheads

b. System throughput

1. A deadlock-avoidance scheme tends to increase the runtime overheads due to the cost of tracking the current resource allocation.

2. However, a deadlock-avoidance scheme allows more concurrent use of resources than deadlock prevention schemes. In that sense, a deadlock avoidance scheme could increase system throughput.

* 1. (5 points) Consider a system consisting of four resources of the same type that are shared by three processes, each of which needs at most two resources. Show that the system is deadlock-free.
     + 1. The system is deadlock-free.
       2. If the system is deadlocked, then each process is holding one resource and is waiting for one more. Since there are three processes and four resources, one process must be able to obtain two resources. This process requires no more resource and can progress. The resources it holds will be return after the process is done. Thus the system cannot be deadlocked.
  2. (5 points) Given five memory partitions of 100 KB, 500 KB, 200 KB, 300 KB, and 600 KB (in order), how would each of the first-fit, best-fit, and worst-fit algorithms place processes of 212 KB, 417 KB, 112 KB, and 426 KB (in order)? Which algorithm makes the most efficient use of memory?

First-fit:

* + - 1. 212KB is put in 500KB partition
      2. 417KB is put in 600KB partition
      3. 112KB is put in 500KB partition
      4. 426KB must wait

Best-fit:

1. 212KB is put in 300KB partition
2. 417KB is put in 500KB partition
3. 112KB is put in 200KB partition
4. 426KB is put in 600KB partition

Worst-fit:

1. 212KB is put in 600KB partition
2. 417KB is put in 500KB partition
3. 112KB is put in 600KB partition
4. 426KB must wait

Best-fit algorithm makes the most efficient use of memory.

* 1. (5 points) Explain the difference between internal and external fragmentation.
     + 1. External Fragmentation occurs when total memory space exist to satisfy a request, but it is not contiguous.
       2. Internal Fragmentation occurs when memory allocated to a process is larger than requested memory. The size difference is memory internal to a partition, but not being used.

1. (35 points) Consider the following snapshot of a system:

|  |  |  |  |
| --- | --- | --- | --- |
| 1) | Allocation | Max | Available |
|  | A B C D E | A B C D E | A B C D E |
| P0 | 1 0 0 1 0 | 4 2 1 2 3 | 1 4 2 1 1 |
| P1 | 2 1 0 1 1 | 2 3 1 6 1 |  |
| P2 | 1 0 1 2 2 | 1 4 2 3 2 |  |
| P3 | 1 1 3 2 0 | 3 6 6 8 4 |  |
| P4 | 3 1 2 1 0 | 5 2 5 2 2 |  |

Answer the following questions using the banker’s algorithm:

1. (8 points) Illustrate that the system is in a safe state by demonstrating an order in which the processes may complete.

Let Need=Max-Allocation

|  |  |  |
| --- | --- | --- |
| Need | Available | Order |
| A B C D E | A B C D E |  |
| 3 2 1 1 3 | 1 4 2 1 1 | P2 |
| 0 2 1 5 0 | 2 4 3 3 3 | P4 |
| 0 4 1 1 0 | 5 5 5 4 3 | P0 |
| 2 5 3 6 4 | 6 5 5 5 3 | P1 |
| 2 1 3 1 2 | 8 6 5 6 4  9 7 8 8 4 | P3 |

We can complete the processes in order of P2, P4, P0, P1, P3.

1. (6 points) If a request from process P1 arrives for (0, 1, 0, 1, 0), can the request be granted immediately?

No, it cannot be granted immediately due to insufficient resource D.

|  |  |  |
| --- | --- | --- |
| Need | Available | Order |
| A B C D E | A B C D E |  |
| 3 2 1 1 3 | 1 4 2 1 1 | P2 |
| 0 ***3*** 1 ***6*** 0 | 2 4 3 3 3 | P4 |
| 0 4 1 1 0 | 5 5 5 4 3 | P0 |
| 2 5 3 6 4 | 6 5 5 5 3 | No Process |
| 2 1 3 1 2 |  |  |

Showed above, it will enter unsafe state, if granted.

1. (6 points) If a request from process P0 arrives for (0, 0, 0, 0, 1), can the request be granted immediately?

No, it cannot be granted immediately due to insufficient resource D, E.

|  |  |  |
| --- | --- | --- |
| Need | Available | Order |
| A B C D E | A B C D E |  |
| 3 2 1 1 ***4*** | 1 4 2 1 1 | P2 |
| 0 2 1 5 0 | 2 4 3 3 3 | P4 |
| 0 4 1 1 0 | 5 5 5 4 3 | No Process |
| 2 5 3 6 4 |  |  |
| 2 1 3 1 2 |  |  |

Showed above, it will enter unsafe state, if granted.

2) (15 points) Consider a system consisting of  resources of the same type, being shared by processes. Resources can be requested and released by processes only one at a time. Show that the system is deadlock free if the following two conditions hold:

a. The maximum need of each process is between 1 and m resources

b. The sum of all maximum needs is less than

Suppose N = Sum of all Needi, A = Sum of all Allocationi, M = Sum of all Maxi. Prove by contradiction.

Assume the system is deadlocked, then A=m because there’s only one kind of resource and resources can be requested and released only one at a time. From condition b, N+A=M<m+n. So we get N+m<m+n, N<n. It shows that at least one process that Need=0. From condition a, the process can release at least 1 resource. So there are n-1 processes sharing m resources now, condition a and b still hold (b changes to “The sum of all maximum needs is less than ”). So, by mathematics induction and n is finite, no process will wait permanently, thus there is no deadlock.

3) (15 points) Memory management

Consider the following segment table:

Segment Base Length

0 220 540

1 2400 600

2 100 90

3 150 60

4 800 200

What are the physical addresses for the following logical addresses?

a. 0,300 b. 1,100 c. 2,500 d. 3,40 e. 4,250

a. 220+300=520

b. 2400+100=2500

c. illegal reference; traps to operating system

d. 150+40=190

e. illegal reference; traps to operating system