ASSUMPTION UNIVERSITY

**Vincent Mary School of Science and Technology**

# **Department of Computer Science**

**Assignment 1 After Midterm Semester** 1/2018

 **Deadlock management**

1. Define the term *deadlock*. Briefly describe the ***four conditions*** that can cause a deadlock in a system.
2. Consider the **resource allocation graph** shown in **Figure 1**(where *R*1, *R*2, and *R*3 are resources with 2 units each and *P*1, *P*2, *P*3, and *P*4 are processes):



 Figure 1

 Attempt the **graph reduction method** and determine whether any ***deadlock*** exists or not (*show all the steps of graph reduction*).

1. Describe the term ‘***safe state***’ in a resource allocation situation.
2. The deadlock avoidance algorithm based on **resource-allocation-graph** is not applicable to a resource allocation system with **multiple instances** of each resource type (as in Figure 1). If so, then suggest and describe a suitable deadlock avoidance algorithm for a system with multiple instances of resources.
3. Consider the following snapshot of a system (where *P*0, *P*1, *P*2, *P*3, *P*4, *A*, *B*, *C*, and *D* are processes, and resources respectively):

 ***Allocation Max Available Need***

 *A B C D A B C D A B C D*

 *P*0 0 0 1 2 0 0 1 2 1 5 2 0

 *P*1 1 0 0 0 1 7 5 0

 *P*2 1 3 5 4 2 3 5 6

 *P*3 0 6 3 2 0 6 5 2

 *P*4 0 0 1 4 0 6 5 6

Answer the following questions using the **banker’s algorithm**:

5.1. What is the content of the matrix ***Need***?

5.2. Is the system in a *safe state*? If yes, then show the safe state sequence.

5.3. If a request from process *P*1 arrives for (0,4,2,0), can the request

 be granted immediately?

**Memory management**

1. Name two differences between **logical** and **physical** addresses.
2. What is *address binding*? Describe the process of *run-time address binding*.
3. Briefly describe the term *dynamic loading*. What is the main advantage of dynamic loading during a program execution?
4. “*Any attempt by a program executing in* ***user mode*** *to access operating-system memory or other users’ memory generates a* ***trap*** (*synchronous interrupt*) *to OS*”. Briefly discuss how would the OS prevent the occurrence of a *trap* during the program execution?
5. Assume that a user process with **100 MB** in size and its backing store is a hard disk with a *transfer rate of* **50 MB** per second. Calculate the ***context-switch time*** of the user process while it’s switching to or from main memory.
6. Describe the terms *external* and *internal* memory fragmentations.
7. What is the purpose of paging the page tables? Describe the advantage(s) of inverted page table over direct page table.
8. A ***segment table*** with five segments and their memory *limit* and *base* values are shown in **Table1**.

|  |  |  |
| --- | --- | --- |
| **Segment no.** | **Limit value** | **Base value** |
| 0 | 1000 | 1500 |
| 1  | 600 | 2600 |
| 2 | 1200 | 3500 |
| 3 | 1100 | 4750 |
| 4 | 400 | 6000 |

Table 1

Based on the segment table, show whether the following references to the byte locations of the segments would result a ***trap*** or not (if not, then show their physical addresses):

10.1. A location reference 1030to**segment 0**

10.2. A location reference 300to**segment1**

10.3. A location reference 1250to**segment 2**

10.4. A location reference 300to**segment 4**

1. What is ***demand paging***? Consider a virtual memory system with a page size of **128-bit** and a physical memory of **256bytes** (organization: 256 ×8). Each logical address has a **4-bit** offset value. The ***page table*** is shown in **Fig 2**.



Figure 2

Based on the **page table**, calculate the ***physical address*es** of the following ***logical addresses***:

11.1. 1216

11.2. 2A16.

11.3. E716.

11.4. F216.

1. What is ***page fault***? Describe various steps taken by OS to handle the page fault.
2. Assume that an *average page-fault service time* of **5 ms** and a *memory access time* of **200 ns**. If one out of 1,000 memory accesses cause a *page fault*, then calculate its ***effective-memory access time*** (**EAT**).
3. Under what circumstances do **page faults** occur? Describe the actions taken by the operating system when a page fault occurs.
4. Assume that a **page reference string** of a user process is “**2 4 2 1 5 2 4 5 3 2 5 2**” (*means that the first page referenced is* 2*, the second page is referenced is* 4*, the third page referenced is* 2*, and so on.*) and a **three** fixed memory-frames are allocated for this process. Show the performance of ***optimal***, ***FIFO***, and ***LRU*** page replacement algorithms in terms of their ***page fault***.