**Worksheets (CH. 7-10)**

**Worksheet #8**

Q1. Draw the wait-for graph for the following situation:

P1 is using R1 and waiting for R2

P2 is using R2 and waiting for R4

P3 is using R5 and R3 and is waiting for R1

P4 is using R4 and is waiting for R5

1. Is there deadlock in this system?
2. If the answer is YES , then which processes are deadlocked

Q2 A system has: 5 processes *P*0 through *P*4; and 3 resource types: *A* (9 instances), *B* (5 instances), and *C* (7 instances). At a time T0 the state of the system is as shown below

***Allocation Max Available***

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

***P*0 1 1 0 6 5 3 2 3 1**

***P*1 1 0 2 3 2 2**

***P*2 2 0 1 80 2**

***P*3 2 1 1 3 2 2**

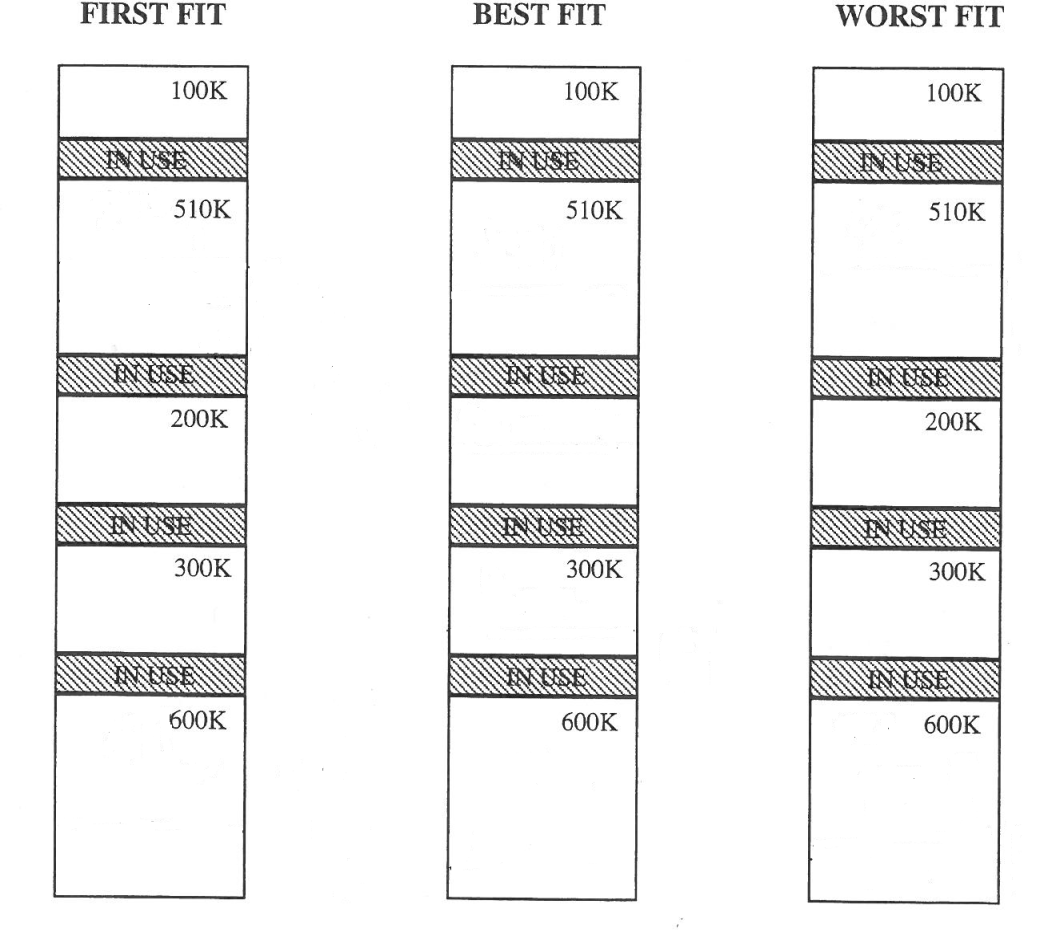
***P*4 1 0 2 4 2 3**

1. Is the system is in a safe state?

1. Can request for (2,2,1) by *P*4 be granted? If YES find the safe state. If NO explain why NOT
2. Can request for (0,2,0) by *P*0 be granted? If YES find the safe state. If NO explain why NOT
3. Can request for (1,0,1) by *P*2 be granted? If YES find the safe. If NO explain why NOT

**Worksheet #9**

Q1. A particular OS manages real memory organized as multiple variable partitions. At the present time there are free memory partitions of 100K, 510K, 200K, 300K and 600K. In the input queue we have several processes waiting to be assigned memory. Process 1 is at the head of this queue and requires 212K, next in line is process 2, which requires 417K, followed by process 3 requiring 112K and then process 4 requiring 426K. Assume that the free storage list is appropriately updated right after process is assigned main memory.



Q3. Consider a paging system with the page table stored in main memory.

1. If access memory reference takes 150 nanoseconds, how long does a paged memory reference take?
2. If we add TLBs, and 55 percent of all page table references are found in the TLBs what is effective memory reference time? (Assume that finding a page table entry in the TLBs takes 5 nanoseconds, if entry is there)

**Worksheet #10**

Q1. Consider a logical address space of 20 pages with 210 words per page, mapped onto a physical memory of 16 frames.

1. How many bits are required in the logical address?
2. How many bits are required in the physical address?

Q2. Assuming a 211 page size, what are the page numbers and offsets for the following address references (provided as decimal numbers):

1. 3454
2. 2014
3. 20572
4. 376
5. 30000
6. 16389

Q3. Consider the following segment table:

|  |  |  |
| --- | --- | --- |
| Segment | Base | Length |
| 0 | 219 | 589 |
| 1 | 2000 | 600 |
| 2 | 1000 | 250 |
| 3 | 1250 | 650 |
| 6 | 3000 | 1200 |

What are the physical addresses for the following logical addresses?

1. 1, 450
2. 0, 367
3. 3, 400
4. 2, 300
5. 6, 1111

Q4. The physical addresses generated by a process can be resolved at compile time, or at load time, or at run time. Which one will best fit each of the memory management schemes named below? Also indicate the type of fragmentation caused by each of the schemes.

1. Multiple Variable Partitions with Compaction

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Multiple Fixed Partitions with a different scheduler for each partition

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Multiple Variable Partitions without Compaction

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Worksheet #11**

Q1. A paging system is experiencing a page fault rate of 1 in 1 million page references. 20% of the time that a page fault occurs, an empty frame is not available and frame replacement is needed. When frame replacement is needed, 60% of time the frame has been modified and it takes 25 milliseconds to service the page fault, in contrast it takes 15 milliseconds when the frame to be replaced has not been modified. Whenever frame replacement is not needed it takes 10 milliseconds to service the page fault. It takes 150 nanoseconds to reference a physical memory location and we can neglect the time that it takes to access the page table.

Calculate the effective memory access time for this system under these conditions. Clearly indicate intermediate steps.

**Worksheet #12**

Q1. Explain the purpose of the open and close operation.

Q2. Consider a file system in which a file can be deleted and its disk space reclaimed while links to that file still exist. What problem may occur if a new file is created in the same storage area or with the same absolute path name? How can these problems be avoided?

Q3. Consistency semantics represent an important criterion for evaluating any file system that support file sharing. Explain the consistency semantics that the Unix file system uses and the Andrew file system uses.

Q4. List and explain the different access methods for files

**Worksheet #13**

Q1. Rank from best to worst, how disk allocation strategies (not including FAT) are suited for implementing sequential file access. How about for implementing direct file access methods?

SEQUENTIAL DIRECT

Q2.What are advantages and disadvantages of contiguous allocation and linked allocation?

Q3. What are advantages of the variant of linked allocation that uses a FAT to chain together the blocks of a file?

**Worksheet #14**

Assume that the following section of main memory is used to store the page table for 3 different processes. The page-table base register values for process P1 is 1080, for P2 is 1085, and for P3 is 1090. Assume that the contests of memory below correspond to frame numbers. Also assume that frame size is 2048.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **contents** | **3** | **5** | **8** | **4** | **7** | **2** | **0** | **15** | **11** | **18** | **6** | **20** | **24** | **9** | **10** | **13** | **30** | **38** | **40** | **1** |
| **Memory addresses** | **1**  **0**  **7**  **6** | **1**  **0**  **7**  **7** | **1**  **0**  **7**  **8** | **1**  **0**  **7**  **9** | **1**  **0**  **8**  **0** | **1**  **0**  **8**  **1** | **1**  **0**  **8**  **2** | **1**  **0**  **8**  **3** | **1**  **0**  **8**  **4** | **1**  **0**  **8**  **5** | **1**  **0**  **8**  **6** | **1**  **0**  **8**  **7** | **1**  **0**  **8**  **8** | **1**  **0**  **8**  **9** | **1**  **0**  **9**  **0** | **1**  **0**  **9**  **1** | **1**  **0**  **9**  **2** | **1**  **0**  **9**  **3** | **1**  **0**  **9**  **4** | **1**  **0**  **9**  **5** |

**To which physical memory address would the logical address (3, 1000) correspond to if generated by P3?**

**Answer:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**To which physical memory address would the logical address (2, 600) correspond to if generated by P1?**

**Answer:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Assume that each process has 5 pages. To which process does the following physical address belong and which logical address corresponds to each physical address:**

**Physical address 14335 Process \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Logical address \_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Physical address 20485 Process \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Logical address \_\_\_\_\_\_\_\_\_\_\_\_\_\_**