Introduction to Computer Science-102

Homework 3

1. What is the advantage of having three transport protocols in TCP/IP? (8%)
   An application can use whichever protocol is appropriate for that application. If an application needs to establish a reliable connection between machines, it can use TCP. If it only needs to send small amounts of noncritical information, it can use UDP to accomplish the transfer faster. SCTP combines the advantages of both UDP and TCP for real-time applications.

2. Why the virtual memory can seem bigger for the physical memory? Please illustrate the concept. (8%)
   This technique virtualizes a computer architecture's various forms of computer data storage (such as random-access memory and disk storage), allowing a program to be designed as though there is only one kind of memory, "virtual" memory, which behaves like directly and contiguous addressable read/write memory.

3. A monoprogramming operating system runs programs that on average need 10 microseconds access to the CPU and 70 microseconds access to the I/O devices. What percentage of time is the CPU idle? (8%)
   $70 / (70 + 10) \times 100 = 87.5\%$.

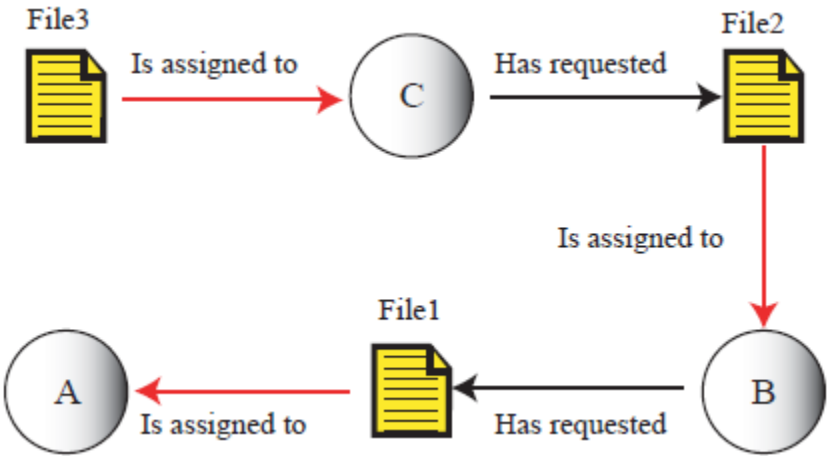
4. A multiprogramming operating system uses paging. The available memory is 60 MB divided into 15 frames, each of 4 MB. The first program needs 13 MB. The second program needs 12 MB. The third program needs 27 MB.(16%)
   a. How many frames are used by the first program?
   b. How many frames are used by the second program?
   c. How many frames are used by the third program?
   d. How many frames are unused?
   e. What is the total memory wasted (not considering memory lost inside each frame)?
   f. What percentage of memory wasted (not considering memory lost inside each frame)?

   a. $13 / 4 = 3.25 \rightarrow 4$ pages.
   b. $12 / 4 = 3$ pages.
   c. $27 / 4 = 6.75 \rightarrow 7$ pages.
   d. $15 - (4 + 3 + 7) = 1$ frame.

e. Not considering memory lost inside each frame, one frame (4 MB) is unused.

f. (4 / 60) × 100 = 6.66%.

5. Three processes (A, B, and C) are running concurrently. Process A has acquired File1. Process B has acquired File2, but needs File 1. Process C has acquired File3, but needs File2. Draw a diagram for these processes. Is this a deadlock situation? If your answer is "no", show how the processes can eventually finish their tasks. (10%)



6. Using the bubble sort algorithm manually sort the following list and show your work in each pass using a table: (10%)

14 7 23 31 40 56 78 9 2

Pass	List								
	14	7	23	31	40	56	78	9	2
1	2	14	7	23	31	40	56	78	9
2	2	7	14	9	23	31	40	56	78
3	2	7	9	14	23	31	40	56	78
4	2	7	9	14	23	31	40	56	78
5	2	7	9	14	23	31	40	56	78
6	2	7	9	14	23	31	40	56	78
7	2	7	9	14	23	31	40	56	78
8	2	7	9	14	23	31	40	56	78

7. A list contains the following elements. The first two elements have been sorted using the selection sort algorithm. What is the value of the elements in the list after three more passes of the selection sort? (10%)

7 8 26 44 13 23 98 57

Pass	List							
	7	8	26	44	13	23	98	57
1	7	8	13	44	26	23	98	57
2	7	8	13	23	26	44	98	57
3	7	8	13	23	26	44	98	57

8. A list contains the following elements. Using the binary search algorithm, trace the steps followed to find 88. At each step, show the value of ***first***, ***last***, and ***mid***. (10%)

8 13 17 26 44 56 88 97

The binary search for this problem follows the table shown below. The target (88) is found at index i = 7

first	*last*	*mid*
1	*8*	*4*
5	*8*	*6*
7	*8*	*7*

1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	
8	13	17	26	44	56	88	97	target > 44
				44	56	88	97	target > 56
						88	97	target = 88

9. Write an algorithm in pseudocode for the selection sort using two nested loops. (10%)

```
Algorithm: SelectionSort(list, n)
Purpose: to sort a list using selection sort method
Pre: Given: A list of numbers
Post: None
Return:
{
    wall ← 1                                  // Set wall at the left of first element
    while (wall < n)                          // Outer loop
    {
        smallest ← wall
        cur ← wall                            // The current item is the one left to the wall
        while (cur < n)                       // Inner loop
        {
            if (A_cur < A_smallest)           smallest ← cur
            cur ← cur + 1                     // Move the current element
        }
        Temp ← A_wall                         // The next three lines perform swapping
        A_wall ← A_smallest
        A_smallest ← Temp
        wall ← wall + 1                       // Move wall one element to the left
    }
}
```

10. Using the insertion sort algorithm manually sort the following list and show your work in each pass using a table: (10%)

14 8 23 60 35 56 71 9 4

14	8	23	60	35	56	71	9	4
8	14	23	60	35	56	71	9	4
8	14	23	60	35	56	71	9	4
8	14	23	60	35	56	71	9	4
8	14	23	35	60	56	71	9	4
8	14	23	35	56	60	71	9	4
8	14	23	35	56	60	71	9	4
8	9	14	23	35	56	60	71	4
4	8	9	14	23	35	56	60	71