

CS 6901 Capstone Exam Systems Fall 2011: Choose any 2 problems.

1) Consider the following two attempted solutions to the 2-process mutual exclusion problem. For each attempt, answer yes/no with a brief justification.

- a) Does the code guarantee mutual exclusion?
 - b) Is it possible that both processes will busy-wait forever?
 - c) Does the code guarantee fairness? That is, is indefinite postponement impossible?
- Briefly explain your answers.

Attempt #1: common variables: flag1, flag2 (both initially false)

```
Process 1                                Process 2
while (true) {                            while (true) {
    flag1 = true;                          flag2 = true;
    while (flag2); //empty body            while (flag1); //empty body
    Critical section;                      Critical section;
    flag1 = false;                         flag2 = false;
    Noncritical section;                   Noncritical section;
}
```

Attempt #2: common variable lock (initially false)

Assume the existence of an indivisible test_and_set function that both returns the value of its boolean argument and sets the argument to true.

```
Process 1                                Process 2
while (true) {                            while (true) {
    while (test_and_set(lock));              while (test_and_set(lock));
    Critical section;                       Critical section;
    lock = false;                           lock = false;
    Noncritical section;                    Noncritical section;
}
```

2) Consider two CPU scheduling algorithms for a single CPU: (nonpreemptive) Shortest-Job-First and Round-Robin. Assume that no time is lost during context switching. Given four processes with arrival times and expected CPU time as listed below, draw a Gantt chart to show when each process executes using

- a) Shortest-Job-First. For this part, also calculate the average turnaround time.
- b) Round-Robin with a time quantum of 4.

Process	Arrival Time	Expected CPU Time
P1	0	7
P2	3	9
P3	5	6
P4	9	4

3) Design a fully simplified 3-bit mod 7 up counter with your choice of JK, D, or T flip-flops. The circuit increments at each clock pulse, going through the sequence 0, 1, 2, 3, 4, 5, 6, 0, 1, 2, Show the circuit diagram.

CS 6901 Capstone Exam Data Structures & Algorithms Fall 2011

Choose any 2 problems. Exam length: 90 minutes

1) Let $a[0], \dots, a[n-1]$ be an array of ints with $a[0] < a[1] < \dots < a[n-1]$. Write a $O(\log n)$ runtime iterative search algorithm that looks for an integer x and returns the index of its first occurrence in the array. Return -1 if it is not found.

2) Write a function which is provided a pointer to the root of a (possibly empty) binary tree, and which returns the integer value corresponding to the height of the tree. Code in the language of your choice and include the declaration of your data structure.

3)

a) Give precise definitions of

i) $f \in O(g)$ (“big-oh”)

ii) $f \in \theta(g)$ (“big-theta”).

b) Let $f(n) = 74n + 41$ and $g(n) = \frac{4}{5}n^2 - \frac{1}{8}n + 1$.

Determine if each of the following 2 statements are true or false. Justify your answers.

i) $f \in O(g)$

ii) $f \in \theta(g)$

Theory Exam

1. Give regular expressions describing each of the following languages over $\Sigma = \{0, 1\}$:

- $\{w : |w| \geq 3 \text{ and the third symbol is } 0\}$
- $\{w : \text{every odd position of } w \text{ is } 1\}$
- $\{w : w \text{ has an odd number of } 1\text{'s and ends with } 0\}$
- $\{w : w \text{ contains at most three } 0\text{'s}\}$
- $\{w : w \text{ starts and ends with the same symbol}\}$

2. Convert the following context-free grammar into an equivalent pushdown automaton:

$$\begin{aligned} S &\rightarrow aSb \mid bY \mid Ya \\ Y &\rightarrow bY \mid aY \mid \varepsilon \end{aligned}$$

3. Let **SUBSET-SUM** = $\{S, t : \text{for some } \{y_1, y_2, \dots, y_l\} \subseteq S = \{x_1, x_2, \dots, x_k\}, \sum_{i=1}^l y_i = t\}$. Show that **3SAT** \leq_p **SUBSET-SUM**.