INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

COMPUTER SCIENCE AND ENGINEERING

Course: CS341 (Operating System), Mid-Semester Exam, Full Marks: 50

Date: 22nd Sept 2014, Time: 2.00PM-4.00PM

1. **[System Architecture, Structure, Service and Design: 5 Lectures (=28%)] 14 Marks[3+4+3+4]**

a) [**3 Marks**] How does OS hide the peculiarities of specific hardware from the users?

b) [**4 Marks**] What are the advantages of using micro-kernel approach in designing an OS?

c) [**3 Marks**] What are the OS service functions that are helpful to users? What are the OS service functions that help in ensuring efficient operation of system itself?

d) [**4 Marks**] What are the differences between user functions, library functions, APIs and system calls? (With examples)

2. **[Process Scheduling and Scheduling Algorithms: 8 Lectures (=44%)] 22 Marks [3+2+6+6+5]**

a. [**3 marks**] What are the differences between short term scheduler and long term scheduler?

b. [**2 Marks**] Given a system with ***n*** processes (arbitrary execution time, pre-emption is not allowed), how many possible ways can those processes be scheduled on a processor?

c. [**6 Marks**] $P_m|out\text{-}tree, p_j=1|C_{max}$ (scheduling ***n*** tasks (*with unit execution time of each task and tasks precedence is out-tree dependency*) on ***m*** homogenous processor system, C_{max}=makespan) can be solved in O(n) time by Hu's Highest Level First (or CF) Algorithm. If we relax $p_j=1$, then execution time of each task will be arbitrary.

Design an efficient approach to solve $P_m|out\text{-}tree|C_{max}$. Your approach may be heuristic or approximation or proper algorithm with time complexity, space complexity and approximation bound (if it is an approximation).

d. [**6 Marks**] Grahm's list scheduling (scheduling of ***N*** independent tasks with arbitrary execution time on ***M*** identical processor) approach approximate C_{max} by a factor of 2. Design an efficient approach to scheduling the same task set on ***M*** uniform processors.

e. [**5 Marks**] Draw Gannt's chart (and calculate average flow time) to schedule the following tasks using FCFS, SJF, SRT, RR (q=2) and RR (q=1) on a processor.

Process	Arrival Time	Processing Time
A	0	3
B	1	6
C	4	4
D	6	2

```
Shared boolean  Lock=false;//free
do {    while(TAS(Lock)) DoNothing();
                 Critical_Section();
          Lock=false;
                     Reminder_Section();
   }          //Text box for Question 3.c
```

3. **[Threading and Synchronization: 5 Lectures (=28%)] 14 Marks [3+3+4+4]**

(a) [**3 Marks**] What are the differences between User Thread, Kernel Thread and Hardware Thread? How threading is beneficial in presence of non-blocking I/Os (async I/Os) and cache miss in processor.

(b) [**3 Marks**] What are the difference between Binary semaphore and counting semaphore? And how semaphore without busy wait can be implemented?

(c) [**4 Marks**] Atomic function/instruction **Test and Set (TAS)** used to implement lock and unlock function of Mutual Exclusion (Mutex). Show that the above (in the Text Box) given TAS solution guarantees Mutual exclusion of a shared variable for M processes on N processors. You don't need PoSet/ToSet/TimedEvent to prove.

(d) [**4 Marks**] What are the assumption Peterson solution assume to prove deadlock free and starvation free? How Peterson solution for Mutual exclusion of two processes on single processor, can be extended to work for two processes on two processors.