Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science
6.826 Principles of Computer Systems

PROBLEM SET 4

Issued: Thursday, February 28, 2002 **Due: Thursday, March 7, 2002**

There are four problems in this problem set; please turn in each problem on a separate sheet of paper and write your name on every sheet. Also give the amount of time you spend on each problem.

Problem 1. Timed Spec

This problem addresses the specification of time in `SPEC`, as explained at page 3 of the Handout 10 on performance.

a) Rewrite procedure `TimedP` from page 3 of the Handout 10 so that it is guaranteed to take *at least* `minLatency` to execute.

b) Show how to initialize `now` and `deadlines` variables and how to define procedure `delay`

```
PROC delay(k: Int) = ...
```

so that in the presence of only one user thread and one `Clock` thread, the following holds:

1. `delay(k)` takes `k` clock ticks to complete
2. time passes only in `delay(k)` actions

c) Are there any problems with your scheme in the presence of multiple user threads?

Problem 2. Optimizing for the Uncommon Case

A just-in-time optimizing compiler knows how to perform two optimizing transformations: A or B. To reduce the time of just-in-time compilation a decision decision has been made to build in the compiler either only transformation A or transformation B.

The performance of the compiler is evaluated on a set of 10 benchmarks each of which is normalized to run for the same time t_0 in the absence of any transformations. We say that a transformation performs better if the average running time of the 10 benchmarks is smaller.

Transformation A reduces the running time of each benchmark 1–9 to $(1 - x)t_0$ where $0 < x < 1$. Transformation A does not change the running time of the benchmark 10.

Transformation B does not change the running time of benchmarks 1-7 but reduces the running time of benchmarks 8-10 to $(1 - y)t_0$ where $0 < y < 1$.

a) For what values of x and y is transformation A better than transformation B, for what values is B better than A and for what values are they equally bad?

b) What is the value of x that guarantees that transformation B can never be better than transformation A? What is the value of y that guarantees that A can never be better than B?

Problem 3. Web Server

Web server requests behave as Poisson arrivals and require two kinds of services: CPU and disk. Requests arrive at average rate n per second and each requires exactly c seconds of CPU time and exactly d seconds of disk time. What is the average response time in which a request gets handled if it has the form:

```
PROC RequestA(data: Data) -> Result =
  res := CPUcompute(data);
  DiskWrite(d,res);
  RET res;
```

and if $c > d$.

Assume that the request needs to wait for the CPU computation to finish before it initiates a disk write. Requests are served in first-come first-served policy.

Problem 4. Widgets

Ben Bitdiddle has quit MIT and started a .com selling widgets to an eager marketplace. His web server is a 800 MIPS machine, with a hard disk bandwidth of 50 MB/s and latency of 6 ms. Each widget transaction requires 200K instructions, and writes a total of 2 KB of data to disk.

a) Ben wishes to balance the speed of the processor with the speed of the disks. How many disks should Ben use to accomplish this if the load on the disks is perfectly balanced?

His e-business online and thriving, Ben modifies his server to batch the transactions' disk I/O in groups of 10. Batched transactions are read and written sequentially to disk.

b) Now how many disks does Ben require to balance the speed of the processor with the speed of the disks?

c) At what rate can now Ben's modified web server handle widget transactions in the steady state?