CMSC 23000	**Operating Systems**	**Homework 3**
Autumn 2006		**Due November 2**

1. Assume that a system is running two reactive processes and one computation-bound process using round-robin scheduling. Furthermore, assume that the reactive processes do a communication operation that blocks for 20ms for every 5ms of computation and assume that it takes 1ms to switch contexts.

   (a) What is the CPU utilization when the scheduling quantim is 5ms?

   (b) What is the CPU utilization when the scheduling quantim is 20ms?

   Note: assume that the preemption occurs at a regular frequency (*i.e.*, the timer does *not* get reset when processes are scheduled.

2. Consider a system with a single-level page table that is kept in physical memory.

   (a) If a physical-memory access takes 30 nanoseconds, how long does it take to access a virtual-memory location?

   (b) If we add a TLB to the system with an access-time of 2 nanoseconds and we assume that 90% of memory references hit the TLB, what is the average memory reference time?

   (c) Suppose that we service TLB misses in software, instead of hardware, and that a TLB miss takes 250 nanoseconds to handle. What is the average memory reference time?

3. Given the following representation of a page table

```
typedef struct {     // A page-table entry
    unsigned    address : 20;     // physical address bits
    unsigned    referenced : 1;   // set when page is referenced
    unsigned    modified : 1;     // set when page is modified
    unsigned    unused : 10;
} pte_t;

typedef struct {     // A page table
    pte_t       *base;            // base address of page table
    pte_t       *top;             // top == base + table size
    pte_t       *next;            // pointer for clock algorithm
} page_table_t;
```

   implement the *clock algorithm* for picking victim pages. Your function should have the following interface:

```
pte_t *pickReplacement (page_table_t *pt);
```

   and should favor replacing unmodified pages over dirty pages.