**BCIT**

**Comp 4735 Operating Systems**

**Instructor: Mirela Gutica**

**Fall 2015**

**Mark: \_\_\_\_\_\_\_\_ / 110**

**Assignment 2**

Note: To receive any credit whatsoever, your answers must be legible and readily readable in the judgement of the grader. Add brief explanatory comments as necessary to make sure your answers are clear and unambiguous to the grader. **When you solve a problem, show all the steps, similar with the examples in the lectures**. Just the answer will not give you credits for a problem. **It is required to have a professional layout for the assignment**.

The assignment should be handed-in no later than **Friday, December 4, 2015, at 11:30pm. No late assignments will be accepted.**

**Solve the problems:**

(20p)

1. Consider a fast food restaurant with cooks, packers and one cashier. Consider that cooks prepare burgers and place them on a table. Packers get burgers from that table, them and place them on a counter for customers to get them. Customers pay first and then are allowed to walk to the table and get burgers.
   1. Define processes and resources.

**Processes**

Cooks, Packers, Cashier, Customers

**Resources**

Burgers

**Other assumptions**

The tables are buffers

The restaurant is always open

The cooks never leave their workstations

* 1. Solve this problem using semaphores (in pseudo-code).

/\* Semaphores & Mutexes \*/

sem\_cooks\_ready = n;

sem\_packers\_ready = n;

sem\_cashier\_ready = 1;

sem\_customer\_ready = 0;

sem\_table\_ready = n;

sem\_table\_occupied = n;

sem\_counter\_occupied = 0;

sem\_counter\_available = n;

mut\_table\_space\_available[] = n;

mut\_packing = 0;

mut\_get\_burger[] = n;

Cook {

while (true) {

makeBurger();

wait(sem\_table\_ready);

wait(mut\_table\_space\_available);

placeBurgerOnTable();

signal(mut\_table\_space\_available);

signal(sem\_table\_occupied);

}

}

Packer {

while(true) {

wait(sem\_table\_occupied);

wait(mut\_packing);

packBurger();

signal(mut\_packing);

signal(sem\_table\_ready);

wait(sem\_counter\_available);

placeBurgerOnCounter();

signal(sem\_counter\_occupied);

}

}

Cashier {

while(true) {

wait(sem\_customer\_ready);

takeOrderFromCustomer();

takeMoney();

signal(cashier\_ready);

}

}

Customer {

goToRestaurant();

signal(sem\_customer\_ready);

wait(sem\_cashier\_ready);

orderBurger();

pay();

wait(sem\_counter\_occupied);

wait(mut\_get\_burger);

getBurger();

signal(mut\_get\_burger);

signal(sem\_counter\_available);

eatBurger();

leave();

}

* 1. Solve this problem with monitors (in pseudo-code).

/\* Semaphores & Mutexes \*/

table\_ready;

table\_occupied;

table\_space\_available;

packing\_space;

counter\_available;

customer\_ready;

cashier\_ready;

counter\_occupied;

get\_burger;

Cook {

while (true) {

makeBurger();

lock (table\_ready) {

lock(table\_space\_available) {

placeBurgerOnTable();

}

}

}

}

Packer {

while(true) {

lock (table\_occupied) {

lock (packing\_space) {

packBurger();

}

}

lock (counter\_available) {

placeBurgerOnCounter();

}

}

}

Cashier {

while(true) {

lock (customer\_ready) {

takeOrderFromCustomer();

takeMoney();

}

}

}

Customer {

goToRestaurant();

lock (cashier\_ready) {

orderBurger();

pay();

}

lock (counter\_occupied) {

lock (get\_burger) {

getBurger();

}

}

eatBurger();

leave();

}

(6p)

1. Consider that you have to design an operating system for critical situations and want to implement a deadlock policy. Indicate three possible solutions and weight the side effects of each.

(8p)

1. Consider a dynamic partitioning system with the following available free blocks (in this order from left to right): 150KB, 360KB, 400KB, 625KB and 200KB. Assume that the memory starts with an occupied block of 100KB and ends with an occupied block of 450KB. The system receives the following requests in this order: P1: 215KB, P2: 171KB, P3: 86KB, and P4: 481KB. (1) Indicate if all new processes can be allocated in the memory and (2) describe the final content of memory indicating the free space for:
   1. Best-fit
   2. First-fit
   3. Next-fit
   4. Worst-fit

Based on this problem and on what we discussed in class, contrast and compare the four placement strategies.

Note: Assume that the last allocation was just before the 400KB block.

(8p)

1. Consider simple paging. Consider that the main memory has 128Mbytes and the size of a page is 2Kbytes. Consider that the memory is byte addressable. A process P has 5 logical pages (first logical page has the address 0).

Answer the questions:

* 1. What is the number of frames in this system?
  2. How many bits are allocated to the page address?
  3. Consider that P has been mapped on frames: 10, 16, 31, 32 and 65 (in this order). What is the physical address for:
     1. 000000000000011110011010011
     2. 000000000000000000111101111
     3. 000000000000010000000000010
     4. 000000000000001001001001001
     5. 000000000000001100000000110

(6p)

1. Explain the advantages of virtual segmentation over virtual paging. Explain why virtual segmentation is not used in modern operating systems.

(6p)

1. A system has a virtual address space of 48 bits and a physical address space of 32 bits. A page is 4Kbytes, the memory is byte addressable, and a page entry has 8Bytes.
2. How many virtual pages are in this system?
3. How many pages are needed for page tables assuming that the page tables are allocated continuously in the memory?
4. Is it possible for the page tables to be stored in the main memory? If the answer is yes, what percentage of the main memory is used by page tables?

(6p)

1. Give as many reasons as you can why locality is a phenomenon that is common during process execution. Give as many reasons as you can of situations when the principle of locality is not happening.

(20p)

1. Consider that the execution of a process requests the following page references:

1, 1, 3, 5, 2, 2, 6, 8, 7, 6, 2, 1, 5, 5, 5, 1, 4, 9, 7, 7

Consider fixed allocation, local scope; 5 frames per process. Show the page allocation and calculate the page fault ratio because page replacement for the following policies:

* 1. FIFO
  2. LRU
  3. Optimal
  4. Simple clock (pre-paging: pages 1,2,3,4,5)
  5. Simple clock (no pre-paging)

(6p)

1. Suppose that a memory management system has chosen a modified page P for replacement. Page P must be sent to secondary storage before the new page is placed in its frame. Therefore, an I/O request is issued. Consider that during the time when the I/O request, the page P is requested again.
   1. What phenomenon describes this situation? Elaborate.
   2. Indicate two strategies that could be employed by the memory management system such that the system executes with an optimal performance? Elaborate.

(24p)

1. Suppose jobs A, B, C, D, and E arrived in a system at time 0, 2, 4, 6, and 7. Assume that the job lengths are: A = 10, B= 5, C = 6, D = 3 E = 5 time units. Draw the time diagrams and calculate Tr and Tr/Ts for each process:
   1. RR q = 1
   2. RR q = 4
   3. SPN
   4. SRT
   5. HRRN
   6. FB, three queues, q1 = 1, q2 = 2, q3 = 4, n1 = 1, n2 = 2.

Note: a. If a process executes its time slice, it cannot be interrupted by the arrival of a new process.

b. If two processes join the queue in the same time: one just finishing the time slice and one new process, the new process has priority over the older one.