74.343 Operating Systems
Paper # 630
Instructors: Eskicioglu and Graham

Final Examination

April 26th, 2002 – 9:00AM
University Centre
220-344

Instructions: Time Allotted - 2 hours.
No aids permitted (or necessary)
Total Marks - 65.

Name: ______________________________

Student Number: ____________________

Question	Value	Score
1	10	
2	8	
3	10	
4	6	
5	8	
6	6	
7	8	
8	9	
	Total:	

[10] 1. Definitions: (Please define *concisely*!)

(i) Counting Semaphore

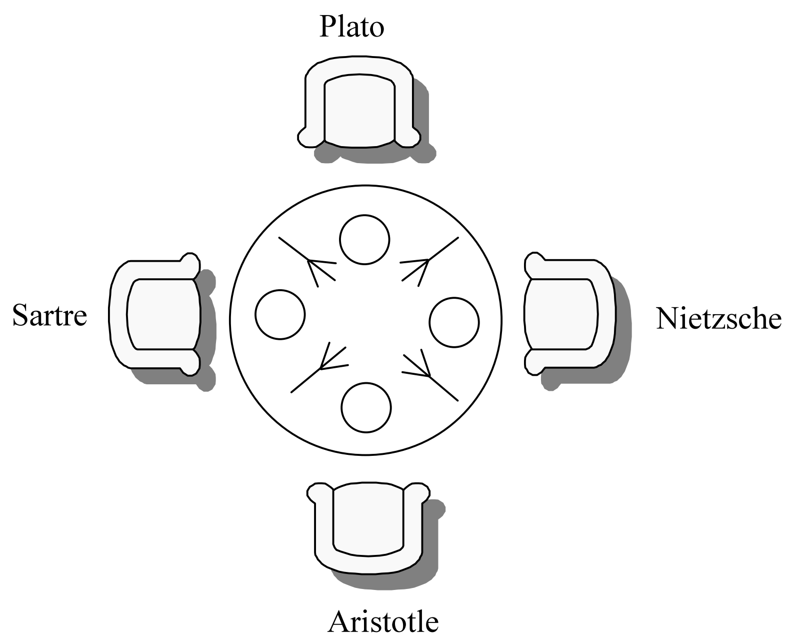
(ii) *Virtual* memory

(iii) Interrupt Handler

(iv) Thrashing

(v) Non-blocking Receive

[8] 2. The “dining philosophers” problem is a classic synchronization problem. Each philosopher has his/her own plate of spaghetti but the forks are shared. There is one fork between each pair of philosophers and a philosopher must have both forks to be able to eat. Each “philosopher” switches between thinking and eating at unpredictable times. When a philosopher is thinking, no forks are required but when a philosopher wants to eat she/he must acquire both forks (the one to the left and the one to the right) before beginning. Once a philosopher has finished eating he/she returns both forks to the table. Pseudo code a solution to the dining philosophers problem using binary semaphores to guard each fork. Specifically, you should pseudo code the routine called “Eat(int p)” which will be executed by each philosopher with the parameter ‘p’ specifying the number of the philosopher invoking the “Eat()” routine. Assume there are N philosophers. Be sure to show your declarations (Hint: use an array of semaphores for the forks). Watch out for deadlocks! A picture of the dining philosophers scenario is provided below for reference:



[10] 3. Consider a paged memory management system that uses the clock algorithm for replacement. There are 5 frames in the memory and the pages initially loaded into them are shown in the first row of the following table. For each page frame there is also a reference bit, as required by the clock algorithm (0 means unreferenced, 1 means referenced). The column labeled "Clock" indicates the number of the frame that the clock pointer points to (frame 3 is where the "clock" points initially). Given the sequence of page references shown in the first column, apply the clock algorithm to complete the table. Be sure to indicate when the faults occur by placing a "Yes" in the rightmost column.

Reference	**Clock**	**Frame 1**		**Frame 2**		**Frame 3**		**Frame 4**		**Frame 5**		**Fault?**
Initial	3	Page 5	1	Page 7	0	Page 3	1	Page 12	0	Page 8	0	***N/A***
Page 7												
Page 5												
Page 4												
Page 10												
Page 12												
Page 2												
Page 12												
Page 9												
Page 17												
Page 9												

[6] 4. A concurrent system consists of 3 processes $\{P_1, P_2, P_3\}$ and 5 resources $\{R_a, R_b, R_c, R_d, R_e\}$. If the allocation sequence: P_2_Alloc(R_b), P_2_Alloc(R_c), P_3_Alloc(R_a), P_2_Alloc(R_e), P_1_Alloc(R_d), P_1_Alloc(R_b), P_3_Alloc(R_c) has already been completed (and assuming all resources were free before the sequence began), draw the full resource allocation graph (RAG) that represents the system.

Is the system deadlocked? Why or why not?

[8] 5. Given a hard disk with 1024 cylinders (C=[0..1023]), 4 heads (H=[0..3]), and 64 sectors (S=[0..63]), fill in the table to show the order in which the sequence of disk I/O requests shown below would be serviced under the NEAREST-FIRST (i.e. seek next to the nearest disk location) algorithm. You should assume that **only** the next **three** requests are available while processing each preceding request. Assume also that the disk read/write heads are originally positioned at cylinder 128 and that all heads move in unison. The requests are in the format: <CYLINDER, HEAD, SECTOR>.

<10,3,42> <200,2,18> <118,1,60> <453,1,6> <1001,2,8> <621,0,0> <8,3,8> <29,2,6>

Current Cylinder	Visible Requests	Selected Request	# Cylinders Crossed
128			
			Σ =

If the cylinder-to-cylinder seek time is 0.1 msec, what is the total disk seek time for all requests?

____

____

[6] 6. Consider two approaches to storing a 1000 block file:

1. Linked List Allocation: There is a pointer to the first data block and each data block contains a pointer to the next.
2. Indexed Allocation: There is a two level index where each block points to 100 other blocks.

For each file system:

a. How many block read operations are required to read **blocks** 500 through 600? Explain!

Linked List: ______________________________

____

____

____

____

Indexed: ______________________________

____

____

____

____

b. How many block read operations are required to read **blocks** 10, 700, 0, 10, 50, and 900 (in that order)? Assume only one request arrives and is processed at a time. Explain you results!

Linked List: ______________________________

____

____

____

____

Indexed: ______________________________

____

____

____

____

[8] 7. Consider the following system snapshot using the data structures in the Banker's algorithm, with resources A, B, C, and D, and processes P_0 to P_4.

	Allocation				Maximum				Available				Need			
	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
									3	3	1	1				
P_0	2	0	0	1	6	3	1	2								
P_1	1	0	0	0	1	7	5	0								
P_2	2	2	5	3	2	3	5	6								
P_3	0	5	3	2	1	6	4	2								
P_4	0	1	1	5	1	6	5	6								

i. How many resources of each type (A, B, C, D) are there? Explain how you know this.

ii. Fill in the current "Need" matrix (above).

iii. Is the system currently in a safe state? Why or why not?

iv. If a request from P0 arrives for additional resources of [1, 3, 1, 0] can the resources be granted? Explain your answer by reference to the system state.

[9] 8. What are the three general *levels* of scheduling discussed in class and where/when is each used?

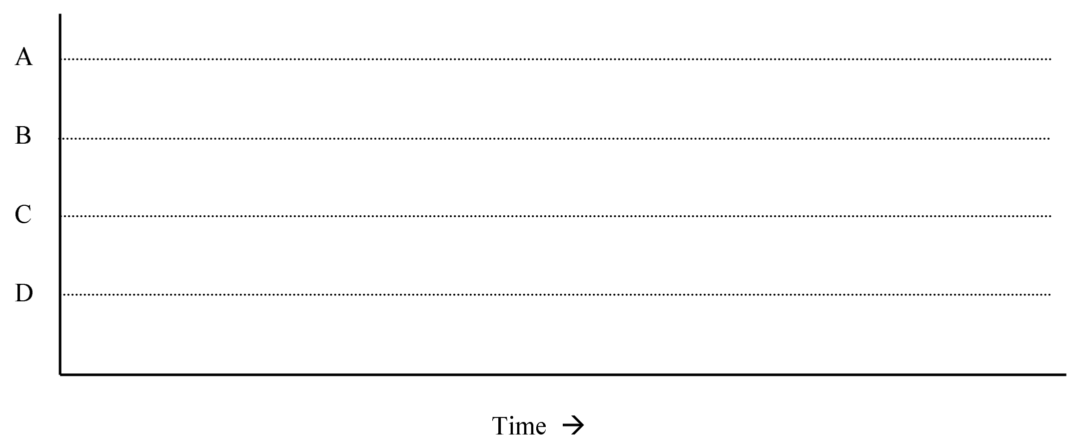
Draw GANTT charts (plots of activity vs. time) for the Round Robin and Priority scheduling strategies for the following jobs (assume larger numbers mean higher priority and that the time quantum is 2):

Job	Arrival Time	Duration	Priority
A	0	6	2
B	2	4	5
C	3	2	4
D	7	3	1

Take note of the arrival times of jobs!

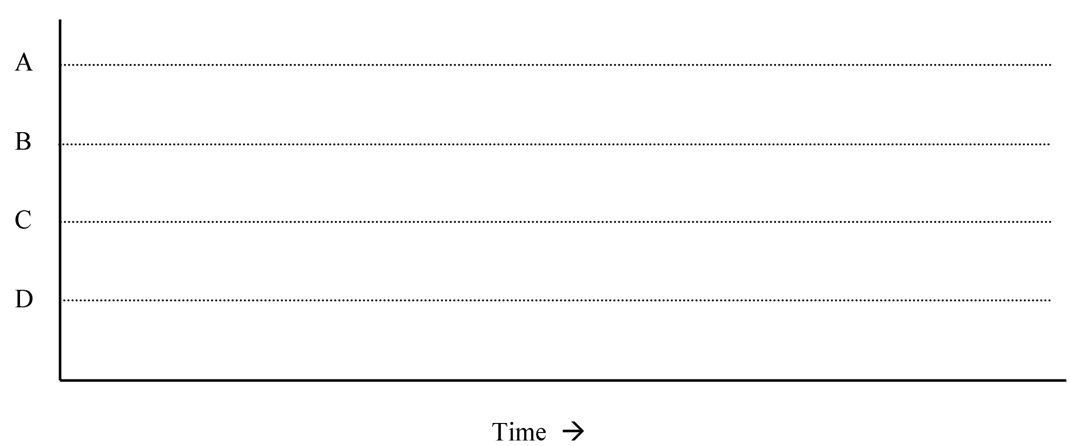
What is the average waiting time under each scheme?

RR



Average wait time for Round Robin is: ______________________________

PRIORITY



Average wait time for Priority is: ______________________________

--- THE END ---