Score:_____ Name:_____

ECE 3055 Test II

1. *(5%)* A virtual memory system has 32-bit virtual addresses and a page size of 2048 bytes. If each entry in the page table is 4 bytes, what is the maximum size of the page table?

$2^{(32-11)} \times 4 = 8,388,608$

Page Table size = 8,388,608 (in bytes)

2. *(5%)* Why does a very small TLB achieve a very high hit rate?

TLB caches pages, pages contain several thousand instructions. This means that accesses to the same page will be a hit. Due to program locality the hit rate will be very high even with a tiny TLB.

3. *(5%)* What is the average time to read or write a sector on a 10,000RPM disk drive with an average seek time of 10ms, a transfer rate of 10.24 MB/sec, and 1024-byte sectors. Assume the disk is idle and there is a 1ms controller overhead per sector.

$10ms + \frac{0.5\ R}{10{,}000\ RPM/60s/m} + \frac{1024B}{10.24\ MB/sec} + 1ms$

$10ms + 3ms + .1ms + 1ms$

Average R/W time per sector = 14.1 ms (in ms)

4. *(5%)* Which type of bus (synchronous or asynchronous) would work best in a system where the bus length and loading could vary widely depending on the system options and configuration? (explain).

asynchronous – handshake lines would handle different delay times with loading and length

5. *(5%)* The latest PCI bus can support 64-bit data and a 66Mhz clock. In burst mode, one clock is required per transfer cycle. What is the maximum possible bandwidth in Mbytes per second?

8bytes/cycle x 66 Mcycles/sec

Bandwidth = 528 (in Mbytes/second)

6. *(5%)* Where would you expect to find queues inside an Operating System?

Ready, Run, Wait queues
Device queues
Semaphore queues

7. *(5%)* What are the tradeoffs in selecting the value of the time slice in a multiprogramming operating system?

Too large - increases turnaround and response time
Too Small - increases time lost in context switching

8. *(5%)* What is the difference between threads and processes?

threads share memory address space
processes have different memory address spaces and therefore require more time for context switches

9. *(5%)* What can you conclude about deadlock in an unsafe state?

deadlock may occur

10. *(5%)* Using the Text's notation, add a semaphore called A to the processes below so that they will function correctly with the shared variable V. V is the only shared variable and all references to V are shown. Semaphore A does not need to be declared in each process.

P0	**P1**	**P2**
.	.	.
.	.	.
.	.	.
WAIT(A);	WAIT(A);	.
IF V=W Then V=X;	**If V=Z then V=Y;**	.
SIGNAL(A);	SIGNAL(A);	.
.	.	.
.	.	.
.	.	.

since V is shared need mutual exclusion around critcal section - Page 168

Set Associative Cache Memory

11. (*10%*)A two-way set associative cache is given the memory operations shown below. It contains eight lines and has one word per block. A LRU (Least Recently Used) replacement policy is used on this cache. The cache is initially empty. The first column of the cache gets the first entry after the valid bits have been reset. In the blanks provided in the Cache below, list the original address value only (i.e. 12 in blank, means address and data for memory location 12 is in cache).

Memory Read Operations

Address	Hit (Y/N)?
2	N
4	N
8	N
5	N
20	N
17	N
19	N
64	N
8	Y
11	N
4	Y
43	N
2	Y
6	N
9	N
917	N

Cache Contents

Block	Element 1	Element 2
0	8	64
1	17	9
2	2	
3	~~8~~ 43	11
4	4	20
5	5	917
6	6	
7		

3/16

Hit rate for all memory operations is 18.75 (%)

Virtual Memory

12. *(15%)* Assume a 16-entry direct-mapped TLB is used in a system with 20-bit virtual byte addresses, 16-bit physical byte addresses, and 256 byte pages. For the sequence of virtual addresses shown below, state whether each address causes a TLB hit or miss and show the physical address generated. Portions of the initial contents of the TLB and full page table are given. Assume all listed page table entries are valid. Show the final contents of the TLB after this sequence of addresses is accessed.

Virtual address sequence

Address	*Hit/Miss*	*Physical Address*
00013	M	2013
022CF	M	A0CF
023A1	M	E2A1
00101	M	F001
000FF	H	20FF
02015	M	7115
02266	H	A066

TLB Initial Contents

Block	*Tag*	*Data*
0	~~02~~ ~~06~~ 02	71 ~~20~~ 71
1	~~02~~ 00	~~C0~~ F0
2	~~00~~ 02	55 ~~A~~ 0
3	~~20~~ 02	E2

TLB Final Contents

Block	*Tag*	*Data*
0	02	71
1	00	F0
2	02	A0
3	02	E2

Page Table Initial Contents

Virtual Page #	*Data*
000	20
001	F0
002	55
003	30
…	
020	71
021	C0
022	A0
023	E2

CPU Scheduling

13. *(15%)* The following processes arrive for execution at the times indicated. Use ~~non~~preemptive scheduling and base all decisions on the information you have at the time the decision is made. At arrival time the execution time is known.

Process	Arrival Time	CPU Execution Time
P1	0.0ms	8ms
P2	0.4ms	4ms
P3	1.0ms	1ms

Draw a Gantt chart using FCFS:

subtract arrival time!

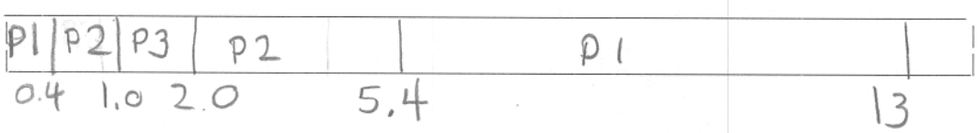
P1 8

P2 12 - .4 = 11.6

P3 13 - 1 = 12

What is the average turnaround time for these processes with FCFS? 10.533

Draw a Gantt chart using SJF:



P1 13

P2 5.4 - .4

P3 2 - 1

What is the average turnaround time for these processes with SJF? 6.33

Deadlock Avoidance

14. *(10%)* The following states are in effect:

Process	Allocation	Max	Need	Available
	ABCD	ABCD	ABCD	ABCD
P0	0 0 1 2	0 0 1 2	0 0 0 0	1 5 2 0
P1	1 0 0 0	1 7 5 0	0 7 5 0	
P2	1 3 5 4	2 3 5 6	1 0 0 2	
P3	0 6 3 2	0 6 5 2	0 0 2 0	
P4	0 0 1 4	0 6 5 6	0 6 4 2	

Process P1 issues a request for (0,4,2,0) can the request be granted?
(justify your answer by showing your work below and show a safe execution sequence, if one exists)

Process	Allocation	Max	Need	Available
	ABCD	ABCD	ABCD	ABCD
P0	0 0 1 2	0 0 1 2	0 0 0 0	1 1 0 0
P1	1 4 2 0	1 7 5 0	0 3 3 0	
P2	1 3 5 4	2 3 5 6	1 0 0 2	
P3	0 6 3 2	0 6 5 2	0 0 2 0	
P4	0 0 1 4	0 6 5 6	0 6 4 2	

P0 Y Avg: 1 1 1 2

P1 N

P2 Y 2 4 6 6

P3 Y 2, 10, 9 8

P4 Y 2 10 10 12

P1 Y 3 14 12 12

Grant Request Y ✓ or N_____

Safe execution sequence = {P0, P2, P3, P4, P1}

or

P0, P2, P1, P3, P4