3

#

COSC 4330 SECOND MIDTERM **MARCH 27, 2002**

This exam is closed book. You can have one page of notes. UH expels cheaters.

- 1. Answer in one sentence to each of the following questions: $(6 \times 5 \text{ points})$
 - What is the major advantage of **user-level threads**? (a)
 - (b) What is the difference between **virtual circuits** and **datagrams**?
 - What is the major limitation of **non-preemptive** scheduling policies? (c)

3

- Why is the **fork**() system call so expensive? (d)
- What is the main reason to use **busy waits** in multiprocessor architectures? (e)
- (f) What is the main disadvantage of **atomic translations**?
- 2. Consider the following System V Release 4 scheduler:

2

100

#ts_quantum	ts_tqexp	ts_slpret	ts_maxwait	ts_lwait	LEV	VEI
1000	0	1	8000	1	# (0
500	0	2	4000	2	# 1	1
200	1	3	1600	3	# :	2

(a) What is the lowest priority level a process can have? (5 points) **Answer: 0**

800

If a process at priority level 2 gets CPU for 100 ms and does a system call, what (b) will be its new priority level when it returns to the ready queue? (5 points)

Answer: 3

3

- Does the scheduler protect all processes from the risk of starvation? (5 points) (c) Justify your answer. (5 points)
- 3. What is wrong with the following solution to the mutual exclusion problem? (10 points to tell me when and how it fails)

```
//Global variables
int tie breaker;
int want_in[2] = {0, 0};
void enter_region(int pid) {
     tie breaker = pid;
     want in[pid] = 1;
     while (want_in[1 - pid] && tie_breaker != pid);
} // enter_region
void leave_region(int pid) {
     want_in[pid] = 0;
} // leave region
```

They do not provide mutual exclusion under almost any circumstance!

4. An interstate bus that can carry 40 passengers, has a single door that let one passenger get in or out at any time. Add semaphores to the following two functions to ensure that the bus will never be overloaded and (b) passengers will not collide with each other when embarking or debarking the bus. (20 points minus 2 points if you forget to initialize the semaphores)

<pre>semaphore passengers =40_;</pre>				
semaphore enter =1_;				
debark(){ embark(){				
	; P(&passengers) ;			
P(&enter);	P(&enter);			
<pre>get_out();</pre>	get_in();			
V(&enter)	;			
V(&passengers)	;;			
} //debark	} //embark			

- 5. What is the Unix system call that processes use to request a connection to another process? (5 easy points)
- 6. How does the VMS scheduler handle processes that return to the ready state after completion of their I/O request? (15 easy points)

SCORE:

COSC 4330

SECOND MIDTERM

MARCH 27, 2002

This exam is **closed book**. You can have **one page** of notes. People caught cheating will **expelled** from UH.

- 1. Answer in one sentence to each of the following questions: $(6 \times 5 \text{ points})$
 - (a) What is the major disadvantage of **user-level threads**?
 - (b) What is the difference between **virtual circuits** and **streams**?
 - (c) What is the major limitation of **UNIX pipes**?
 - (d) What is the main disadvantage of **busy waits** in uniprocessor architectures?
 - (e) What is the main advantage of **atomic translations**?
 - (f) Why is the **fork**() system call so expensive?
- 2. Consider the following System V Release 4 scheduler:

#ts	_quantum	ts_tqexp	ts_slpret	ts_maxwait	ts_]	wai	t]	LEVE]	Ĺ
	1000	0	1	8000		0	#	0	
	500	0	2	4000		2	#	1	
	200	1	3	1600		3	#	2	
	100	2	3	800		3	#	3	

- (a) What is the highest priority level a process can have? (5 points) **Answer:**
- (b) If a process at priority level 1 waits for 1200 ms in the ready queue and has the CPU for 200 ms before returning to the ready queue, what will be its new priority?
 (5 points)
 Answer: _____
- (c) Does the scheduler protect all processes from the risk of starvation? (5 points) Justify your answer. (5 points)
- **3.** What is wrong with the following solution to the mutual exclusion problem? (10 points to tell me when and how it fails)

```
//Global variables
int can_go;
int request[2] = {0, 0};
void enter_region(int pid) {
    can_go = pid; // set tiebreaker
    request[pid] = 1;
    while (request[1 - pid] && can_go != pid);
} // enter_region
void leave_region(int pid) {
    request[pid] = 0;
} // leave_region
```

4. An interstate bus that can carry 50 passengers, has a single door that let one passenger get in or out at any time. Add semaphores to the following two functions to ensure that the bus will never be overloaded and (b) passengers will not collide with each other when embarking or debarking the bus. (20 points minus 2 points if you forget to initialize the semaphores)

semaphore bus =;	
<pre>semaphore door =;</pre>	
embark(){	debark(){
;	;
;	;
get_in();	get_out();
;	;;
;	;;
} //embark	} //debark

- 5. Give one example of **private ports** in the UNIX system. (5 **easy** points)
- 6. How does the VMS scheduler handle processes that return to the ready state after completion of their I/O request? (15 easy points)