

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

1. *Advantages and disadvantages:* (4×5 points)

- a) What is the major disadvantage of *not using lightweight processes in a server*?

The server will either have to remain single-threaded or will have to fork conventional processes, which would be much costlier than creating a lightweight process.

- b) What is the major disadvantage of *busy waits*?

They waste CPU cycles (and context-switches).

- c) What is the major advantage of *streams* over *datagrams*?

They guarantee that all messages will arrive in sequence, undamaged, without lost or duplicate messages.

- d) What is the major advantage of *idempotent procedures*?

A remote procedure call to an idempotent procedure can be restarted as many times as needed until it executes properly.

2. What is the default action taken by a process receiving a *signal*? (5 points) How can the process specify a different outcome? (5 points)

By default, a process that receives a signal terminates. To specify a different outcome, the process can do a `signal()` system call to catch the signal.

3. Consider the following system V.4 scheduler.

#ts_quantum	ts_tqexp	ts_slpret	ts_maxwait	ts_lwait	LEVEL
800	X	1	16000	1	# 0
400	0	2	8000	1	# 1
200	1	3	4000	Y	# 2
100	2	Z	2000	3	# 3

- a) The sole correct value for the parameter **X** is: 0

- b) The sole correct value for the parameter **Y** is: 3

- c) The sole correct value for the parameter **Z** is: 3

4. For each of the statements below, indicate in one sentence whether the statement is true or false (2 points), and why (3 points).

a) Servers should always use **blocking** receives.

TRUE, there are more efficient than non-blocking receives.

b) Most modern operating systems use **preemptive scheduling**.

TRUE, preemptive scheduling guarantees that high-priority processes will never have to wait for a lower-priority process that monopolizes the CPU.

c) We can avoid process starvation by **increasing** the priorities of processes doing terminal I/O.

FALSE, we should increase the priorities of the processes that have been in the ready queue for too long.

d) **Atomic transactions** guarantee **at most once semantics**.

FALSE, atomic transactions guarantee all-or-nothing semantics.

e) A message sent to a **private mailbox** can only be received by one of the processes having read access to that mailbox.

FALSE, a message sent to a private mailbox can only be received by the process owning the mailbox or its children.

f) In a RPC, one of the tasks of the **user stub** is to exchange messages with the user program.

FALSE, the user stub exchanges messages with the server stub.

5. How could we achieve fast context switches between threads sharing the same address space without requiring programmers to use non-blocking system calls? (10 points)

We could do what Amoeba does. Amoeba threads are managed at user-level but the kernel is aware of them: when a thread executes a blocking system call, the kernel returns control to the thread scheduler of the task; this thread scheduler can either schedule another thread or return control to the OS.

6. Complete the following program segment so that the output of **stdout** will be redirected to the pipe **outpipe**: (3×5 points)

```
int outpipe[2];
_pipe(outpipe)_____ ;
close(_1_____)
dup(____outpipe[1]_____) ;
close(outpipe[0]); close(outpipe[1]);
```