# Exam Revision

Edward Zhang

SOFTENG 370 T8

# Exam Info

Your exam will be short answer, not MCQ. That means the exam from 2012 - 2017 aren't very useful. 2018 had a different lecturer for the first $\frac{1}{4}$ so it's not super helpful either.

## Which of the following is not a necessary component of a monitor?

- Publicly accessible entry points
- A readers/writers lock
- A scheduler
- A shared resource which is protected by the monitor

# Which of the following is not a necessary component of a monitor?

- Publicly accessible entry points
- A readers/writers lock
- A scheduler
- A shared resource which is protected by the monitor

## Explanation

Reader/Writers lock can enhance performance, but is not required.

## Which of the following best explains what happens when a damaged C program comes to an end but doesn't call the exit routine?

- The damaged program can corrupt memory used by other processes and cause them to crash or perform illegal instructions.
- The operating system takes control when the program tries to execute an illegal instruction or attempts to access unallocated memory.
- The C standard library takes control when the program fails to return to the code which called the main function.
- The operating system creates a new process and restarts the damaged program in that process so that it gets another chance to complete.

## Which of the following best explains what happens when a damaged C program comes to an end but doesn't call the exit routine?

- The damaged program can corrupt memory used by other processes and cause them to crash or perform illegal instructions.
- The operating system takes control when the program tries to execute an illegal instruction or attempts to access unallocated memory.
- The C standard library takes control when the program fails to return to the code which called the main function.
- The operating system creates a new process and restarts the damaged program in that process so that it gets another chance to complete.

# The code below uses a compare and swap function “cas”. What is the code doing?

```
add_to_balance(increase):
  previous_amount = balance
  while (!cas(&balance,
    previous_amount,
    previous_amount + increase)):
  previous_amount = balance
```

- It repeatedly increments balance by increase until balance overflows.
- It increments balance by increase using a condition variable.
- It safely swaps the values of balance with previous_amount + balance using a wait-free algorithm.
- It safely increments balance by increase using a lock-free algorithm.

# The code below uses a compare and swap function "cas". What is the code doing?

```
add_to_balance(increase):
  previous_amount = balance
  while (!cas(&balance,
    previous_amount,
    previous_amount + increase)):
  previous_amount = balance
```

- It repeatedly increments balance by increase until balance overflows.
- It increments balance by increase using a condition variable.
- It safely swaps the values of balance with previous_amount + balance using a wait-free algorithm.
- It safely increments balance by increase using a lock-free algorithm.

## Which of the following does NOT happen in a context switch between threads in the same process?

- The processor registers for the currently running thread are saved.
- The processor registers are loaded with the saved values for the new thread.
- The page table is switched from the old thread to the new thread.
- The thread states for the two threads may be changed.
- The stack is changed from the old thread to the new thread.

# Which of the following does NOT happen in a context switch between threads in the same process?

- The processor registers for the currently running thread are saved.
- The processor registers are loaded with the saved values for the new thread.
- The page table is switched from the old thread to the new thread.
- The thread states for the two threads may be changed.
- The stack is changed from the old thread to the new thread.

## Explanation

Memory is shared between threads, so same page table.

## Which of the following is False?

- FUSE works by redirecting file operations through the FUSE module to a process running in user mode.
- To use a FUSE file system we mount the file system over an existing directory.
- To use FUSE requires root privileges.
- If the FUSE process is killed the files and directories contained within it will not be accessible.
- There has to be a FUSE kernel module in order for FUSE to work on Linux.

# Which of the following is False?

- FUSE works by redirecting file operations through the FUSE module to a process running in user mode.
- To use a FUSE file system we mount the file system over an existing directory.
- To use FUSE requires root privileges.
- If the FUSE process is killed the files and directories contained within it will not be accessible.
- There has to be a FUSE kernel module in order for FUSE to work on Linux.

## Explanation

You probably used FUSE w/o root in your assignment.