CS 350 Notes

Matthew Visser

Oct 4, 2011

1 Assignment Issues

1.1 Errors in A1

If you detect an error, you can

- assert
- panic
- fail silently
- put return values in the lock struct.

In other settings you might be able to change return values, but since the API is define dfor us, we can't.

1.2 `sy2` Test

Will run the lock test. This test passing once does not guarantee that your lock implementation is correct.

2 System Calls

There are 3 ways to enter the kernel:

- syscalls
- interrupts
- exceptions

On MIPS, these all jump to `0x80000080`.

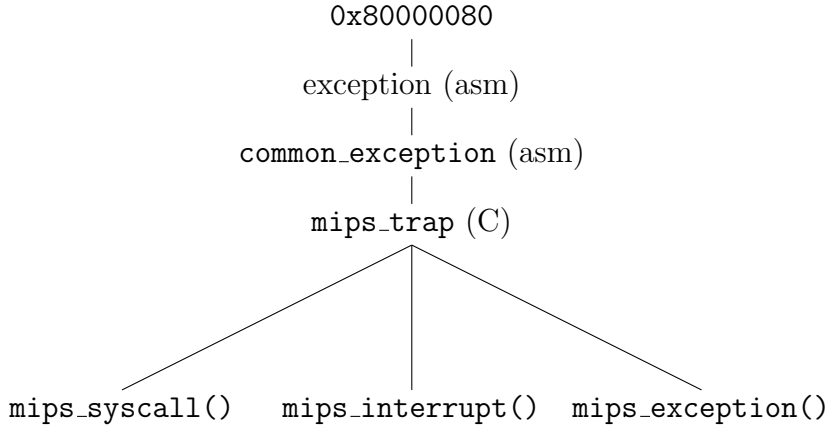
2.1 The Exception Handler

Every thread has two stacks, one in user mode and one in kernel mode. The thread library points to the kernel stack.

The exception handler does the following:

1. allocates a trap fram on thread's kernel stack and saves user-level application's state.
2. calls `mips_trap`
3. when this returns, restores application and processor state
4. issues MIPS `jr` and `rfe` instructions to return control to application.

The calling code can either call an exception, interrupt, or syscall.



A syscall returns an error code and an optional return value. Pass a pointer to the return value variable to the syscall functions to be able to change it. This is OK because nothing other than `mips_syscall()` will be calling it.

3 Exceptions

- Conditions that occur during execution of user code
- Detected by hardware.
- hardware calls exception handler
- examples are
  - page faults

– divide by zero

3.1 Alignment

When we load or store, we need to load or store from an alignment that's divisible by the size of the register. Unaligned memory causes exceptions.