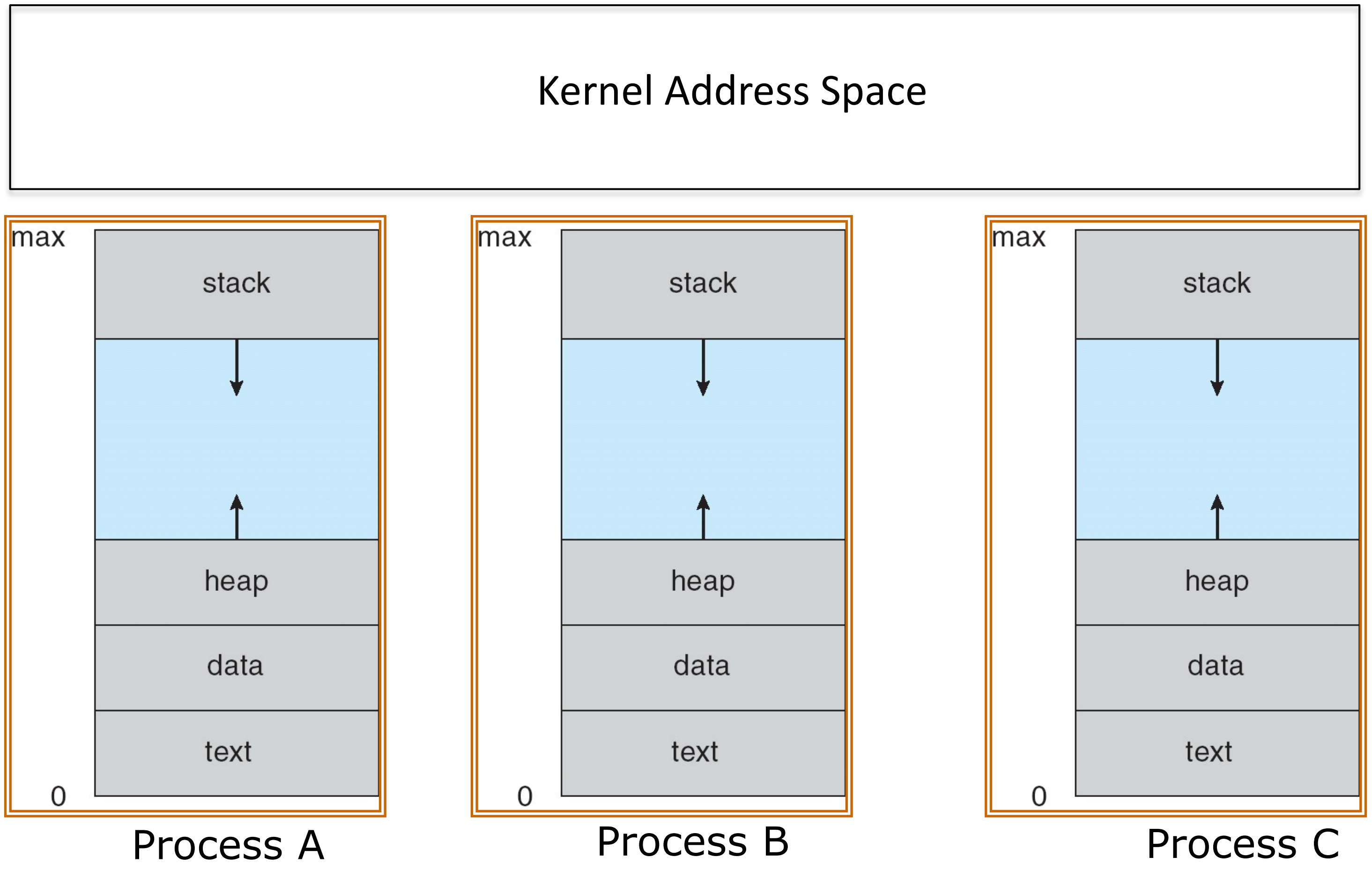
System Calls

COMS W4118

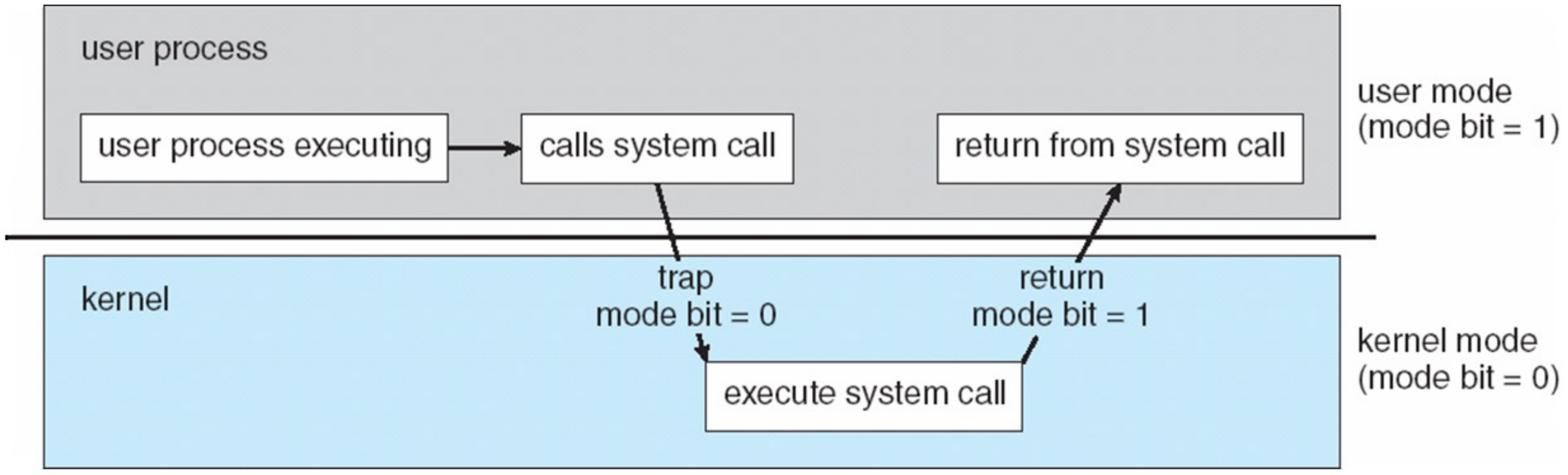
References: Operating Systems Concepts (9e), Linux Kernel Development, previous W4118s

Address Space Overview



System calls

- User process normally runs in unprivileged **user mode**
  - Cannot perform privileged operations
- User process issues **system call** to enter **kernel mode**
  - Privilege elevated, but only for predefined functions

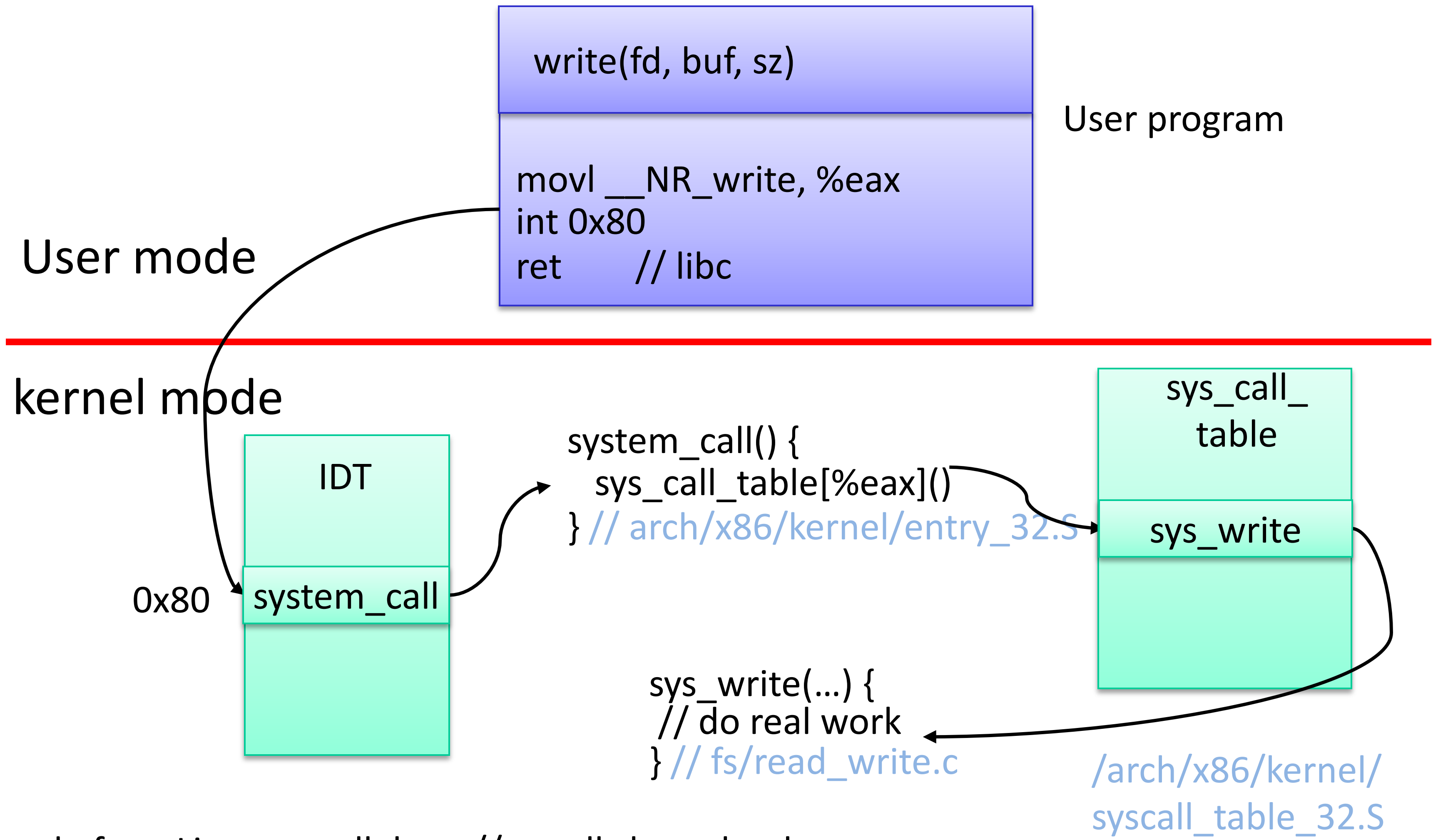


Three kinds of interrupts

- Hardware interrupts
  - Ex) network packet, timer, key press, mouse click
- Exceptions
  - Ex) dividing by zero
- Software interrupts
  - Ex) int 0x80

```
while (1) {
    if (interrupt or exception) {
        n = interrupt/exception type
        call interrupt handler n
    }
    fetch next instruction
    if (instruction == int n)
        call interrupt handler n
    else
        run instruction
}
```

Linux System Call Dispatch



To see code for a Linux syscall: http://syscalls.kernelgrok.com

Linux System Call Parameters

- Syscall parameters are passed in registers
  - Max size of argument is register size
  - Larger argument passed as pointer
- `copy_from_user()/copy_to_user()`
  - Validate all pointer arguments to syscalls
  - May block if memory has been swapped to disk
  - So you cannot be holding spin locks

Tracing system calls in Linux

- Use the “strace” command (man strace for info)
- Linux has a powerful mechanism for tracing system call execution for a compiled application
- Output is printed for each system call as it is executed, including parameters and return codes
- ptrace() system call is used to implement strace
  - Also used by debuggers (breakpoint, singlestep, etc)
- Use the “ltrace” command to trace dynamically loaded library calls