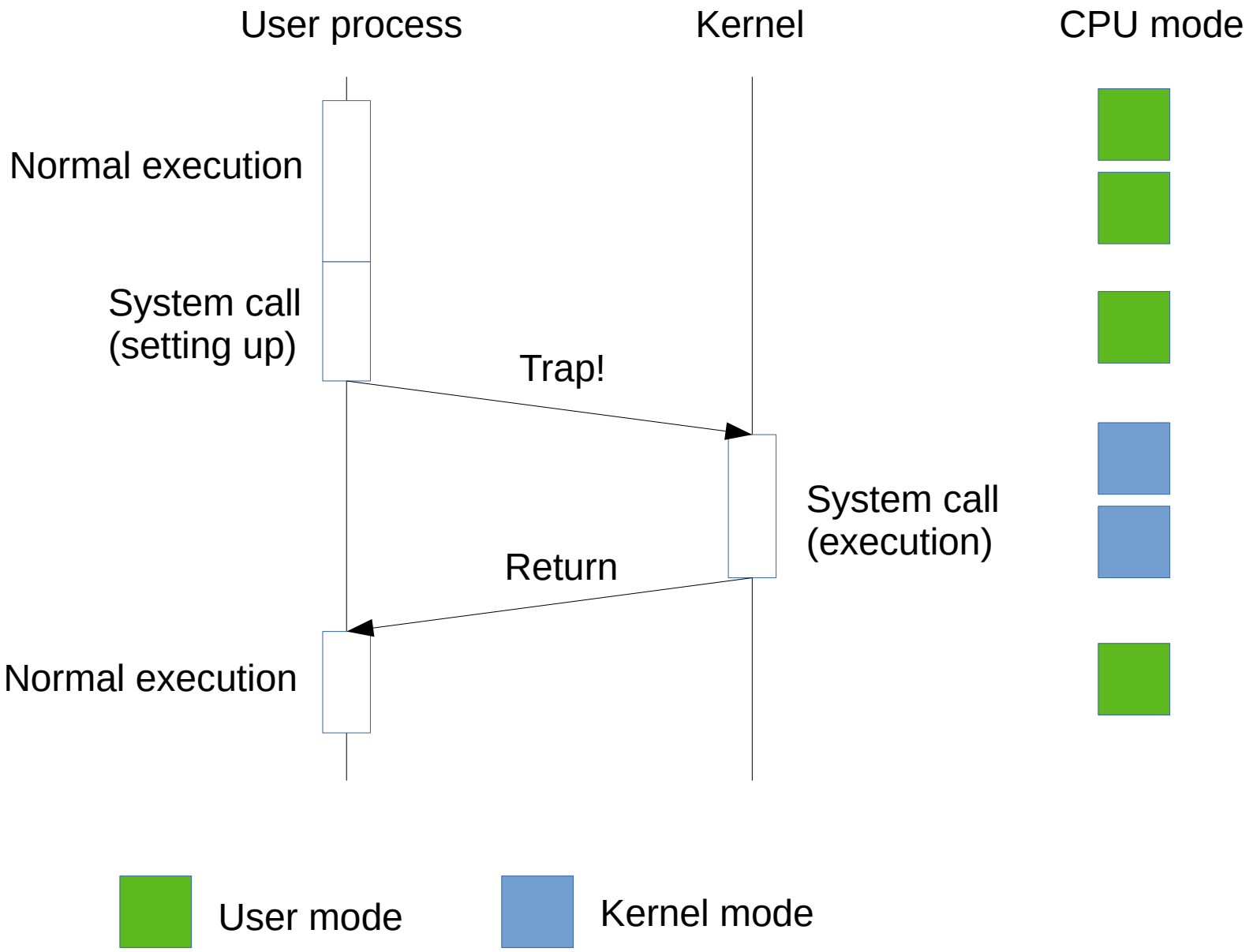
TDDB68/TDDE47
Concurrent Programming and Operating Systems

Lecture 1, part 5:
Dual mode and system calls

Mikael Asplund

Dual mode


User process
Kernel
CPU mode
Normal execution
System call
(setting up)
Trap!
System call
(execution)
Return
Normal execution
User mode
Kernel mode

Types of System Calls

- **Process control**
  load, execute, end, abort, create, terminate, wait ...
  memory allocation and deallocation
- **File management**
  open, close, create, delete, read, write, get/set attributes...
- **Device management**
  request / release device, read, write, ...
- **Information maintenance**
  get / set time, date, system data, process / file attributes
- **Communications**
  create / delete connection, send, receive, ...

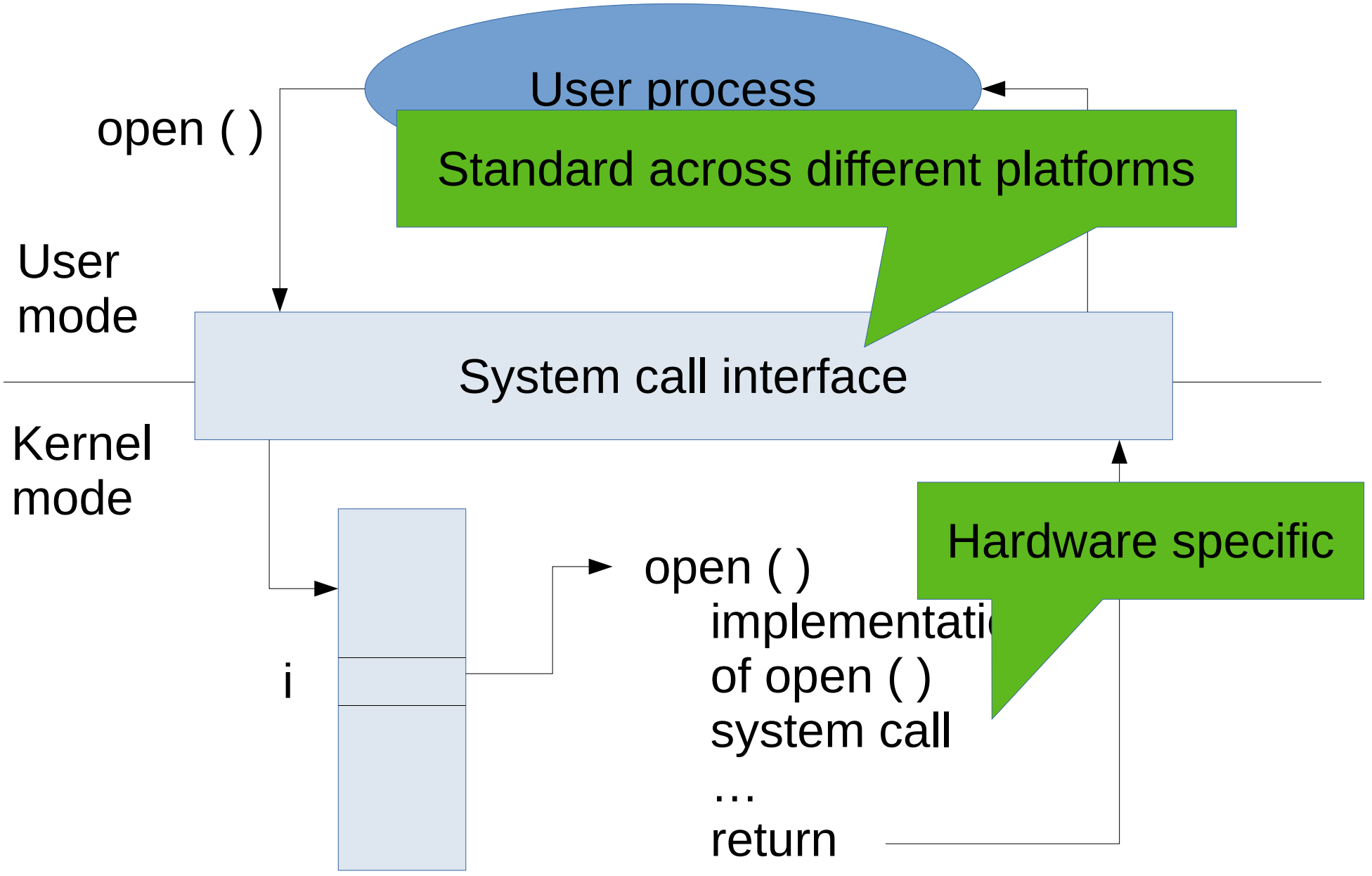
Syscalls In Linux

(man syscalls)

Tracing system calls

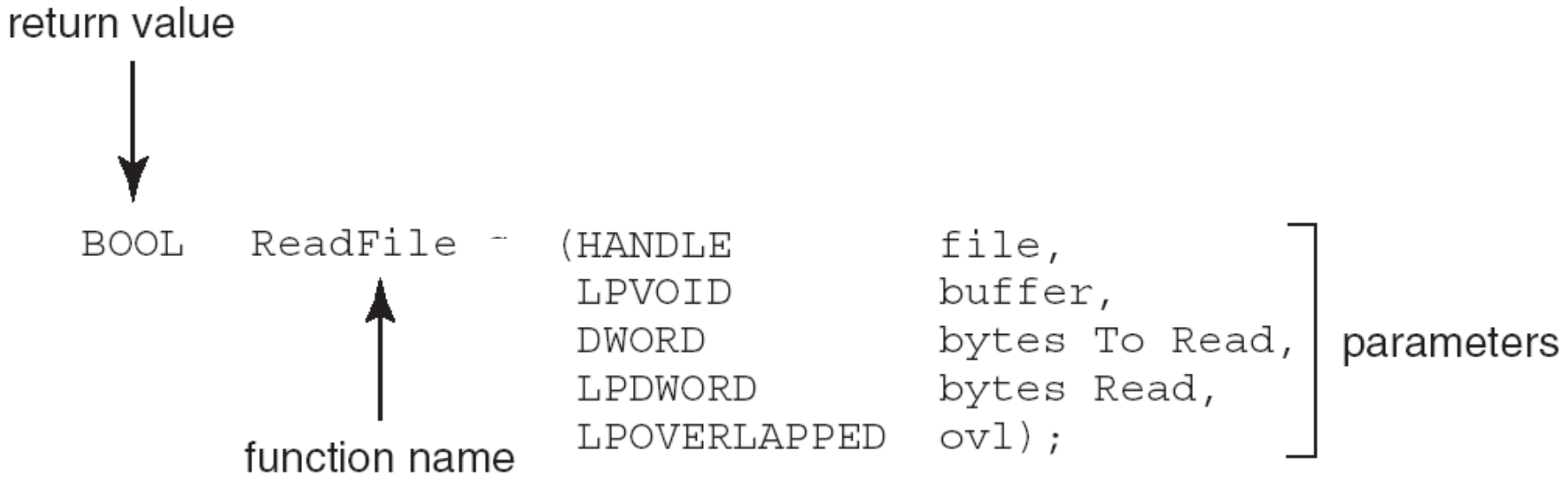
Linux – ltrace
Mac OSX - dtrace

System Call API – OS Relationship



Example of a System Call API

- ReadFile() function in Win32 API (function for reading from a file)



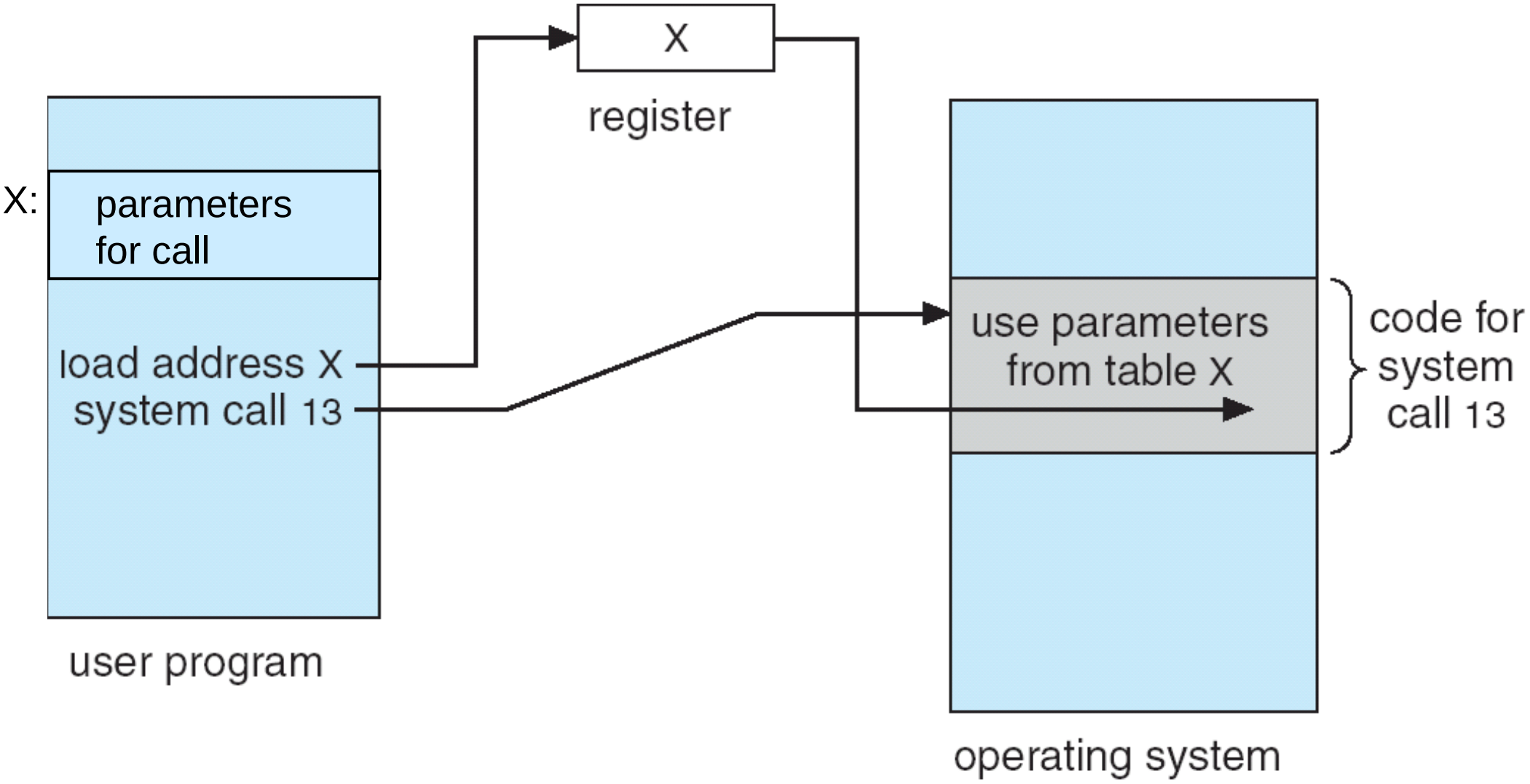
- Parameters passed to ReadFile():
  - file — the file to be read
  - buffer — a buffer where the data will be read into and written from
  - bytesToRead — the number of bytes to be read into the buffer
  - bytesRead — the number of bytes read during the last read
  - ovl — indicates if overlapped I/O is being used

System Call Parameter Passing

- Three general methods used to pass parameters to syscalls:
  - Parameters **in *registers***
  - Parameters **in a *block in memory***, and address to block in a register
    - This approach taken by Linux
  - Parameters ***pushed onto the stack***

- What are the advantages/disadvantages?

System Call Parameter Passing via Block



Summary

- **Operating System** = OS Kernel + System Programs
  - Mediates all accesses to system resources
  - **Interrupt**-driven
    - Error handling
    - Controlled access to system resources, e.g.
      - I/O devices, DMA
      - CPU time sharing
    - …
- **Dual-Mode** (user mode, kernel mode)
  - **System Call** API for portability

Reading guidelines

- 9th edition
  - Chapter 1: Sections 1.1-1.7
  - Chapter 2: 2.3-2.5
- 10th edition
  - Chapter 1: 1.1-1.5
  - Chapter 2: 2.3-2.4

What's next?

- **Monday**
  - Sign up in Webreg!
  - Get the book
- **Tuesday**
  - 13-17 Introductory C lecture
  - Try some C programming
- **Thursday**
  - 8-10 lesson to introduce the labs
- **Friday**
  - First lab (lab0)