CS503: Final Exam Information
Spring 2014

Department of Computer Science
Purdue University

Date/Time/Location

- Thursday, May 8, 2014
- 8:00AM – 10:00AM (morning exam!)
- ME 1130

Scope

- Chapters 1-9 of Xinu book
- Chapters 12, 13, 14, 15, 19 of Xinu book
- All lecture notes I covered in class
- 30% before mid-term
- 70% after mid-term

Types of Questions

- 10-15 short "Q&A" problems
- One problem on virtual memory, page replacement, working set (not Xinu-specific)
- One problem on device driver
- One problem on file system (UNIX and Xinu)
- One open problem

OS Abstraction and Application Interface (from Prof. Comer's notes)

- API
  - Defines uniform interface
  - Provides access to abstractions
  - Hides ugly hardware
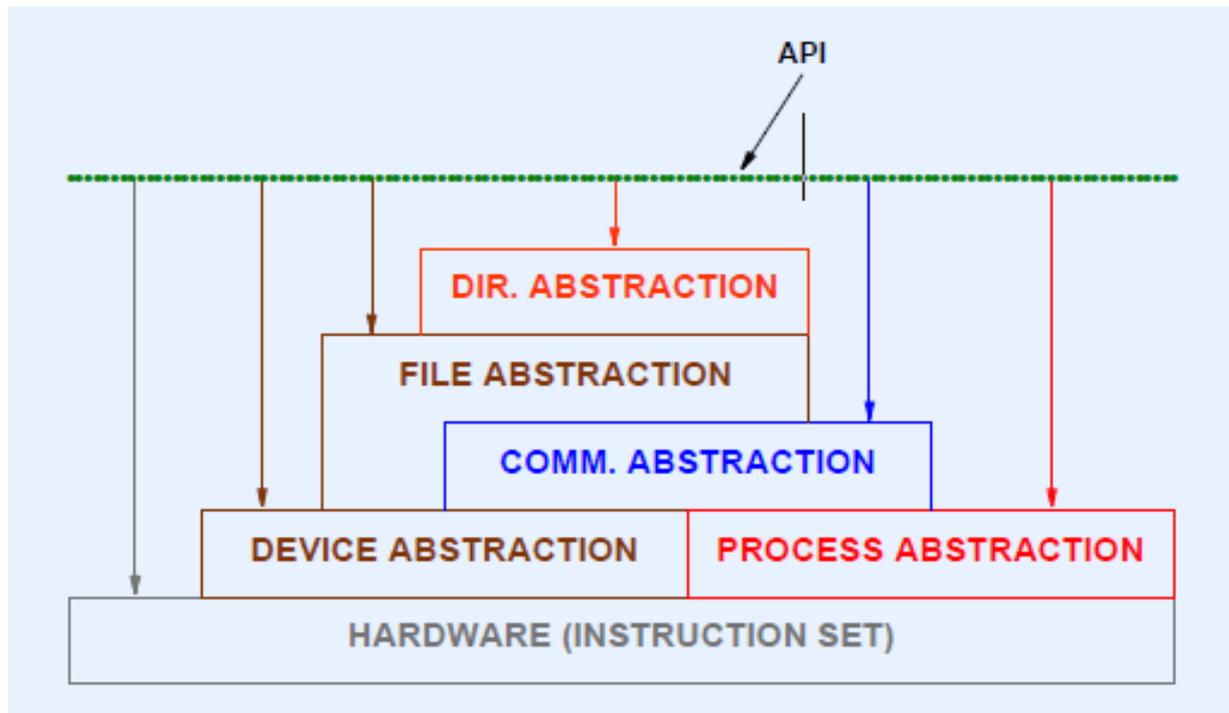
What You Learned

- In-depth understanding of major parts of an operating system, including:
  - Process manager
  - Memory manager
  - Interrupt processing and device manager
  - Interprocess communication
  - File manager
- A hierarchical approach to organizing the pieces
- The division into policy, design, and implementation
- Appreciation of design decisions and tradeoffs
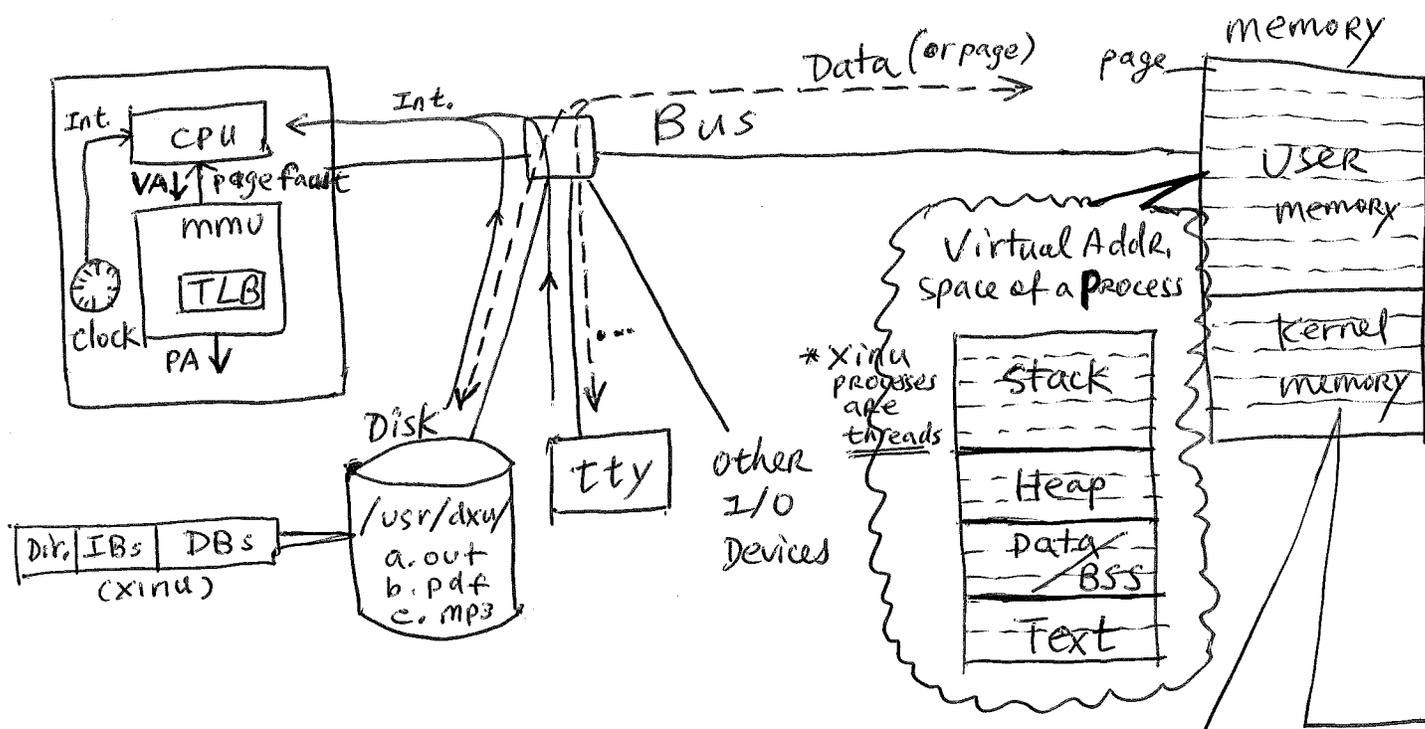- A key set of invariants and principles

6

What You Learned (Cont.)

- How an OS can provide convenient abstractions over low-level hardware

- How OS functions can be implemented without resorting to millions of lines of code

- How seemingly straightforward policies can interact in unexpected ways

- How intellectually challenging it can be to understand all the consequences of even a basic system
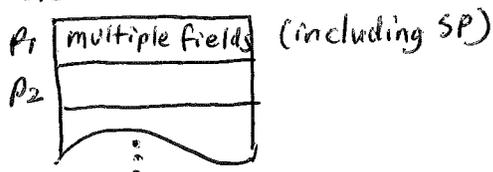
After This Course

- You either
  - Will not have to touch OS internals. But I hope you will still remember the "big picture" 5 (and 10, 15…) years down the road
  - Will have to explore even further. I hope this course has "ushered" you into the area of OS (and other system software) design and implementation
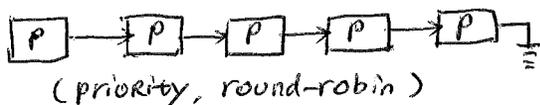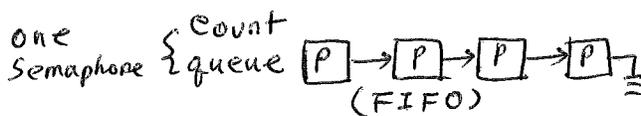
CS503 in 20 minutes

Data (or page) → Bus

page → memory

CPU — Int. — Int.

VA ↓ ↑ page fault

mmu

TLB

clock

PA ↓

Disk

/usr/dxw/
a.out
b.pdf
c.mp3

| Dir. | IBs | DBs |

(xinu)

tty

other
I/O
Devices

Virtual Addr.
Space of a Process

*Xinu
processes
are
threads

Stack
Heap
Data / BSS
Text

USER
memory

kernel
memory

• **Process table** (resched, create, suspend ...)

P_1 | multiple fields | (including SP)
P_2 |

• **Ready Queue** (resched, ctxsw ...)
Ready,

P → P → P → P → P

(priority, round-robin)

• **Semaphores** (wait, signal ...)

one
Semaphore { Count
queue P → P → P → P

(FIFO)

• **Delta List** (sleep, wakeup ...)

K P → K P → --- → K P

• **Page Table** (one per process)

| V | R | D | phy. pg. # |

} subset
cached
at
TLB
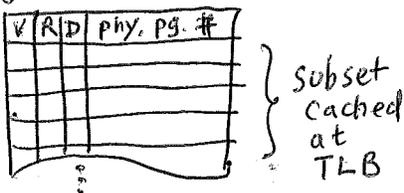
• **Mem list for**
low-level memory
management

memlist → □ → □ → □
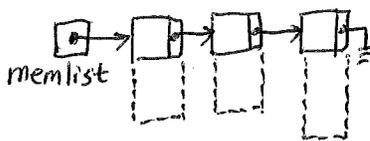
• **Device Switch Table**

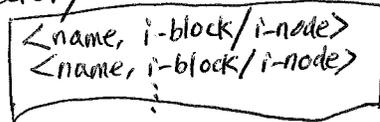|	open	close	read	write	getc	putc
tty						
Ether						
rds						

upper
half

Device
Driver

Shared
data
structures

ctrl blk | ctrl blk | ctrl blk | ctrl blk

• **Interrupt vector** (intdispatch.s
dispatch.c ...)

clock → clkhandle
tty → tty Interrupt
Ether → Eth Interrupt
local disk → Disk Interrupt

lower
half

• **Directory**

<name, i-block/i-node>
<name, i-block/i-node>

• **I-node** (UNIX)

File Info

• **I-block** (Xinu)

Dir.
IB → IB → IB

DB