

Outline

- **Last time**
 - More intro, History (Chapters 1 and 2)
 - Assignment #1 out (later today, due Thurs Sept 13 11am)
- **This time**
 - Assignment #1 overview
 - Hardware review (Chapt 2)
 - OS Structures (Chapt 3)
- **Next time (Tonight 6-7:15pm OLS 120/ Tomorrow 3-4:15 OLS 120)**
 - Finish up HW concerns (Chapt 2)
 - More Chapter 3: OS structures

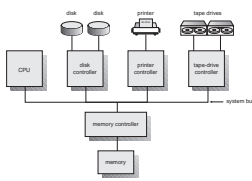
CS414: Operating Systems

Before we start

- **My Office Hours:**
 - Monday 1-4pm, Tues 10-11am or by appointment
- **Graduate TA Office Hours:**
 - Blake Sheridan (sheridan at cs.virginia.edu), Wed 5-8pm 002a
- **Other TA**
 - Ming Mao (mm5bw at cs.virginia.edu)
- **Email questions? Please send to all three. One or more will reply ASAP.**
 - Don't expect email from me between 9pm and 5am

CS414: Operating Systems

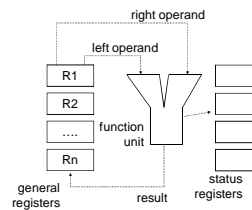
Computer Architecture



- 1940s: von Neumann et. al.: defined specific organization for stored program computers (as opposed to single-purpose electronic devices)
- **Process:** Intuitively, corresponds to sequential execution of a program on a set of data on a von Neumann architecture
- Instructions and data must reside in primary memory before they are to be used by a process
 - A defining characteristic of the von Neumann architecture

CS414: Operating Systems

Inside the CPU: ALU

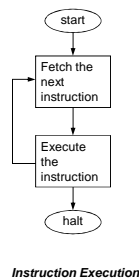


- Typically, CPU has 32-64 general registers
 - each register can hold 32 bits (perhaps 64/128)
- General registers are sometimes referred to as user-visible registers
 - they can be read from/written to by machine language
- **Functional unit**
 - add, multiply, shift, logical AND, etc., on binary representations of ints and floats
- **Usually, there is a special floating point unit performing functionality on floats**
- "Output" of ALU is often used in special register program status word (PSW)
 - can include: Sign, Zero, Carry, Equal, Overflow, interrupt enable/disable, Supervisor

CS414: Operating Systems

Inside the CPU: Control Unit

- **Control Unit has components for:**
 - fetching next instruction
 - decoding instruction
 - signaling other parts of the computer to execute the instruction
- **Most current CPUs perform fetch and decode (and perhaps execution) in parallel**
- (at least) **Two fundamental registers: Program Counter (PC) and Instruction Register (IR)**
- ROM usually contains small **bootstrap loader**



CS414: Operating Systems

Pipelining

- **Pipelining: overlap instruction execution to reduce the total time to complete an instruction sequence.**
 - Not every instruction depends on immediate predecessor \Rightarrow executing instructions completely/partially in parallel is possible
 - Classic 5-stage pipeline:
 - 1) Instruction Fetch (Ifetch),
 - 2) Register Read (Reg),
 - 3) Execute (ALU),
 - 4) Data Memory Access (Dmem),
 - 5) Register Write (Reg)
- **Hazards prevent next instruction from executing during its designated clock cycle**
 - **Structural hazards:** attempt to use the same hardware to do two different things at once
 - **Data hazards:** Instruction depends on result of prior instruction still in the pipeline
 - **Control hazards:** Caused by delay between the fetching of instructions and decisions about changes in control flow (branches and jumps).
- **Can always resolve hazards by waiting**

CS414: Operating Systems

ISA

•**ISA: Instruction Set Architecture: data types, instructions, registers, addressing modes, memory architecture, interrupt and exception handling, external I/O**

- Opcodes
- Microarchitecture: implementation of the ISA (e.g., AMD and Intel both implement x86)
- CISC vs. RISC
- Examples: MIPS, x86, IA-64, Alpha AXP, VAX, PowerPC

•**OS (with a few exceptions) largely only utilizes/engages/manages the Virtual Machine that by definition is exposed via the ISA**

- E.g., OS does not generally re-order instructions (dynamically)
- Compiled source code + some ASM = OS

CS414: Operating Systems

Performance

•**Cycles Per Instruction (CPI): relates the Instruction Set Architecture (ISA) with the implementation of that architecture**

•**Amdahl's law:**

- Make the common case fast
- Speedup is limited by the UNIMPROVED part of the program

CS414: Operating Systems

Amdahl's Law example

•**New CPU 10X faster**

•**I/O bound server, so 60% time waiting for I/O**

$$\text{Speedup}_{\text{overall}} = \frac{1}{(1 - \text{Fraction}_{\text{enhanced}}) + \frac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}}}$$
$$= \frac{1}{(1 - 0.4) + \frac{0.4}{10}} = \frac{1}{0.64} = 1.56$$

•**Hmmm... not 10X faster but just 1.6x faster**

CS414: Operating Systems

Improvements in technology

•**Moore's law: 1965 : # transistors per sq inch doubles every 18 months (or 24 months)**

- However, we cannot continue to increase operating frequency (e.g., heat)

•**Simultaneous multithreading (e.g., P4 Hyperthreading)**

- "improve apps 20% by using 5% of the real-estate of the chip"
- Works by duplicating certain sections of the processor but not duplicating the main execution resources
- When one task is stalled (e.g., cache miss, branch misprediction, data dependency), execute another task
- Often appears "TO" the OS as being multiple processors
 - OS must already have support for symmetric multiprocessing (SMP – single shared memory)

CS414: Operating Systems

Improvements in technology

•**Multi-cores...**

- Non-shared L1 cache and shared L2 cache
- Memory, not ALUs, is the bottleneck
 - RAM access times and HW seek times improve a "few percentages" each year
 - Multi-cores makes this worse (!)
- Apps will need to be multi-threaded
 - Hey! Let's have a thread assignment in this class! Cool!
- Games successful already

•**But beware the "memory wall"! (see Amdahl's law)**

•**Niklaus Wirth's law (1995): "Software gets slower faster than hardware gets faster"**

CS414: Operating Systems

Memory

•**Memory interface: memory address register (MAR), memory data register (MDR), command register (CMD)**

•**Access time to memory is called memory cycle, which is usually 1-4 basic clock cycles**

•**Generally, memory is byte-addressable, although the data bus transfers words (2 or 4 bytes)**

CS414: Operating Systems

CS414 Programming Assignments

- **“Here’s the Hardware; come back when you have an OS”**
 - Compelling, but too challenging
- **“Here’s Java; investigate scheduling, synchronization, etc.”**
 - Compelling, but perhaps too “far away from HW”
- **“Here’s NACHOS”**
 - Compelling, but not exactly what I want
- **“Here’s Linux; investigate scheduling, synchronization, etc.”**
 - Compelling...
- **“Here’s the Windows kernel; investigate scheduling, synchronization, etc., in a VM environment”**
 - Compelling... let’s go with it and Linux – a broad mix!

CS414: Operating Systems

WRK (Windows Research Kernel)

Source from the latest shipping Windows (NTOS) kernel

- Version – Windows Server 2003 (x86/x64) and Windows XP x64
- Included sources – most everything in NTOS – processes, threads, LPC, VM, scheduler, object manager, I/O manager, synchronization, worker threads, kernel memory manager, ...
- Excluded sources – plug-and-play, power-management, and specialized code such as the driver verifier, splash screen, branding, timebomb, etc.
- Build environment – makefile-based with object library for the excluded sources; Kernels boot on native hardware or using VirtualPC

Microsoft

WRK Goals

- Simplified licensing to allow classroom and lab use
- Make it easier for faculty and students to compare and contrast Windows to other operating systems
- Enable students to study source, and modify and build projects
- Provide better support for research and publications based on Windows internals
- Encourage more OS textbook and university-oriented internals books on Windows kernel

Microsoft

WRK Licensing

- **Goals**
 - Faculty feel comfortable agreeing to its conditions
 - Students can use in classroom/lab environment
- **License type**
 - Non-commercial, academic use only; allows derivative works for non-commercial purpose
- **Eligibility criteria**
 - Available to faculty and students in colleges/universities WW
- **Usage scenarios**
 - View, copy, reproduce, distribute within the institution
 - Modify for teaching and experimentation purposes
 - Produce teaching and research publications including relevant snippets of source
 - Can use in textbooks and academic publications, and community forums
 - Have to perpetuate Microsoft copyright notices
 - Share derivatives within academic community

Microsoft

HW #1

•HW #1 Out now (due Thurs 9/13)

	PRO	CON
002a	More already pre-installed	No local storage, potential for losing stuff upon reboot, have to leave your couch
Your machine	Easier access	Not enough CPU/HD/RAM?

- Part 1: Exploring the WRK
- Part 2: Compiling WRK
- Part 3: Running Win Server 2003 as the VM
- Part 4: Running your WRK

CS414: Operating Systems