Overview

- This Lecture
  - Introduction
  - Reference:
- Next Lecture
  - Network hardware
  - Reference: *Ethernet: The Definitive Guide*, Charles E. Spurgeon, O'Reilly

CS engineering cliff

- CS students understand less and less computer systems
  - Blame Java!
- Hands-on experience on computer systems is valuable
  - The industry needs more graduates understanding computer systems
  - Internet of Things, wearable computer, wireless sensors
  - System-on-chip (SoC), cheap DIY electronics

Network management

- What's it about?
  - Originated from system administration, especially for multi-user systems
    - User/accounting management
    - System/software installation and configuration
    - System maintenance
  - Computer networks and the Internet have broadened the area of system administration
    - Network maintenance/monitoring
    - Network service installation and configuration
    - Security : hosts, network, data
    - Cloud management

Network management (cont.)

- System/network administration as a profession
  - It was not a discipline, just passed on by word of mouth. But now there is a LISA, Large Installation System Administration forum and other conferences.
  - It is about putting together a network of computers, getting them running, and then keeping them running in spite of the activities of users and crackers who tend to cause the systems to fail.
  - System/network administrators need a broad range of knowledge and skills. They are services providers, mechanics, sociologists, psychologists, doctors, ...
  - As a system/network administrator, no one thanks you for what you have done but blame you when anything goes wrong.

Objectives of the course

- Sound understanding of computer systems and networks from hands-on experience
  - You should understand what you are doing. It is required for stability and security of your systems.
  - IT grads requirements: system integration, scripting, fixing simple PC problems.
  - Do you know how to change Linux scheduler behaviour?
- Understand and practise every aspect of system and network administration
  - Focus on principles/theories. What you do today may be different from tomorrow, but principles survive longer.
  - You should be able to propose new principles based on your experience and expertise.
  - Management theory may help.

Objectives of the course (cont.)

- Focus on the technical aspect of system and network management
  - The technical skills, not the administrative skills.
- Use Linux plus Ethernet plus VM as a case study
  - You will feel confident to maintain a small network of computers based on Linux and Ethernet and can extend your knowledge to other networks and systems.
  - Linux know-how is topping most highly sought expertise http://www.eweek.com/c/a/IT-Management/Linux-Job-Openings-on-the-Rise-Dice-Report-600529/.
- A place to test your practical knowledge
  - If you feel you don't understand something, you either lack the knowledge or haven't associated it with what you have learnt before. Do ask questions!

Resources

- The textbook: The Lab Handbook (use the e-copy online) plus the lecture notes
- Recommended textbooks (reserved in the library)
  - Mark Burgess, *Principles of Network and System Administration (2nd Edition)*, John Wiley & Sons
  - T. Bautts, T. Dawson, and G. Purdy, *Linux Network Administrators Guide (3rd Edition)*, O'Reilly
  - Bob Toxen, *Real World Linux Security—Intrusion Prevention, Detection, and Recovery*, Prentice Hall PTR
  - Limoncelli and Hogan, *The Practice of System and Network Administration (2nd Edition)*, Addison Wesley
  - Tom Limoncelli, *Time Management for System Administrators*, O'Reilly
  - Brendan Gregg, *Systems Performance*, Prentice Hall

Resources

- Related materials from **The Linux Document Project http://en.tldp.org**
  - GNU/Linux Command-Line Tools Summary
  - Linux System Administrators Guide
  - Linux Network Administrators Guide
  - Bash Guide for Beginners
- Links for system administrators
  - Everything Sysadmin website: http://everythingsysadmin.com
  - The Operations Report Card: http://www.opsreportcard.com
  - SAGE Code of Ethics: https://www.usenix.org/lisa/system-administrators-code-ethics

Resources (cont.)

- Linux manual pages
  - Use **man** and **apropos** commands
- Related web sites
  - The Linux Document Project, **http://en.tldp.org**
    - HOWTO documents
  - Linux web site, **http://www.linux.com/**
  - Ubuntu site, **http://www.ubuntu.com/,** check the archive of the mailing lists for any problems.
  - Linux kernel, **http://www.kernel.org/, http://kernelnewbies.org/**
  - Administrator's forums and conferences such as USENIX, SAGE, and LISA, **http://www.usenix.org/, http://www.sage.org/**
  - Linux conferences such as linux.conf.au
- **Maybe the above resources are obsolete, the last but most important resort is Google search!**

Course Details

- Nature of the course
  - No absolute truth.
  - Steadily changing area.
  - De-valued if no hands-on experience.
- How we'll teach it
  - Provide coaching rather than teaching in labs.
  - Lab intensive.
  - Lectures focus on understanding rather than simple recipes.
  - Encourage learning from errors.
  - Encourage team work and interest-driven adventure.

Course Details (cont.)

- Assessment
  - 45% for assessed labs each of which worth 1% or 2%, or even 5%.
  - Assessed labs are divided into three parts, each part has a due date.
  - 5% practical test in week 9.
  - 50% for final examination. Note you have to pass 40% of the final exam.
- Important points
  - Go to lectures and labs.
  - This course is very hierarchical. If you miss labs or lectures you may not be able to follow the rest.
  - Good relationship is a must. This paper is a chance to practise team work in the course as administrators do.
  - Read the lab material before you go to a lab.

Notice

- http://www.cs.otago.ac.nz/cosc301/
- Lecture notes will not be printed. Get them from the course website before each lecture.
- Do The Prelab!
  - Before Wednesday
  - Get the *Lab Handbook* by clicking the **Lab Manual** tab at the above course website
- The COSC301 lab is at Lab F in Owheo Building
  - Email Kaye Saunders kaye@cs.otago.ac.nz if you have lab stream clashes
  - See Paul Crane for any other lab problems

System administrator

- Successful Administrators
  - No stereotypes. “White/black Cats Theory”
  - Normally if one can keep the users happy he/she is successful (which is very difficult).
  - Life-long learning skills. One’s proud knowledge and expertise today may be obsolete tomorrow (Be prepared to be jobless if you do not keep learning).
- Some myths for new SAs
  - There exists a right answer for every problem.
  - Things should always work in the way we expect.
  - Every problem should have a happy end.

System administrator (cont.)

- Challenges of SAs
  - Not just installing system/software, also about planning and designing an efficient community of computers.
  - Design a logical and efficient network.
  - Easy upgrade for a large number of computers.
  - Decide what and where services are installed.
  - Plan and implement security.
  - Provide a comfortable/convenient environment for users.
  - Develop ways of fixing problems and errors.
  - Efficient administrative skills.
  - Keep track of new technology and software.

Practice of SAs

- Common practice may not be good practice
  - There are three reasons for common practice:
    - Someone did it and others followed blindly;
    - Believe it is good after careful thought;
    - An arbitrary choice had to be made;
  - Think for yourself.
  - Pay attention to experts but don't automatically believe anyone.
  - Every choice needs a reason, which is why you need to learn this paper.

Practice of SAs (cont.)

- Good practices
  - Look for answers in manuals, newsgroups, and archive of mailing lists. Usually "google" helps for most common problems.
  - Use controlled trial and error for diagnosis.
  - Listen to people who tell us there is a problem. It might be true.
  - Write down problems and solutions in a log book, and write down experiences (highly recommended).
  - Take responsibilities for our actions.
  - Remember to tidy things up regularly.
  - After learning something new, ask yourself *"How does this apply to my work?"*

Superuser

- Superuser (root)
  - What is root (superuser)?
  - Be aware of the double-edged sword
    - Convenient to do anything
    - Powerful enough to damage the system
- Login as superuser (root)
  - System admins should never login as root
    - Many commands can be executed by ordinary users
  - When you need root privilege
    - Use **sudo** or **sudo -s** or **su**

Linux commands/syscalls today

- > man apropos
- > apropos scheduler
- > which nice
- > nice
- sched_setscheduler()
- nice(), getpriority(), setpriority()
- capget(), capset()

Overview of Operating System

Computer Architecture

- Computer architecture

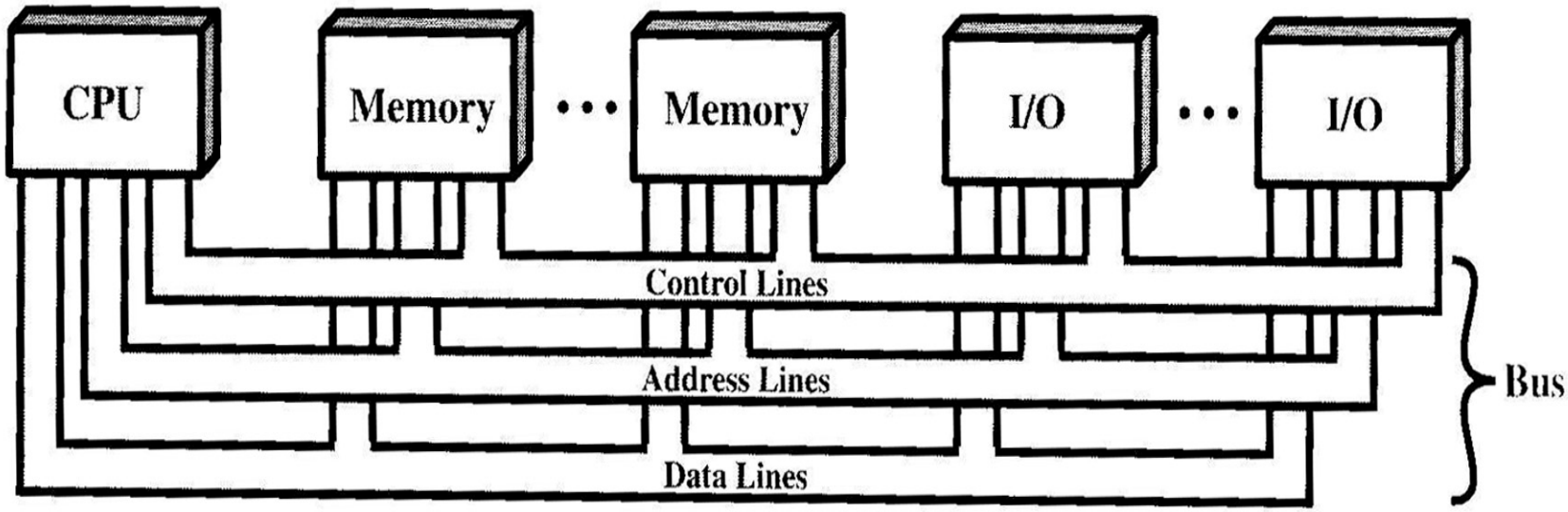


Figure 3.16 Bus Interconnection Scheme.

Computer Architecture (cont.)

- CPU
  - does arithmetic and logic operations
  - Executes instructions
- Memory
  - stores instructions and data, e.g. programs
  - How does CPU access memory?
- I/O devices
  - e.g. hard disks, monitor, CD-ROM
  - have a control card communicating with CPU through buses (ISA, PCI, USB)
  - How does CPU control I/O devices?
  - Interrupts/traps and registers
- How does a program get started?

Origin of OS

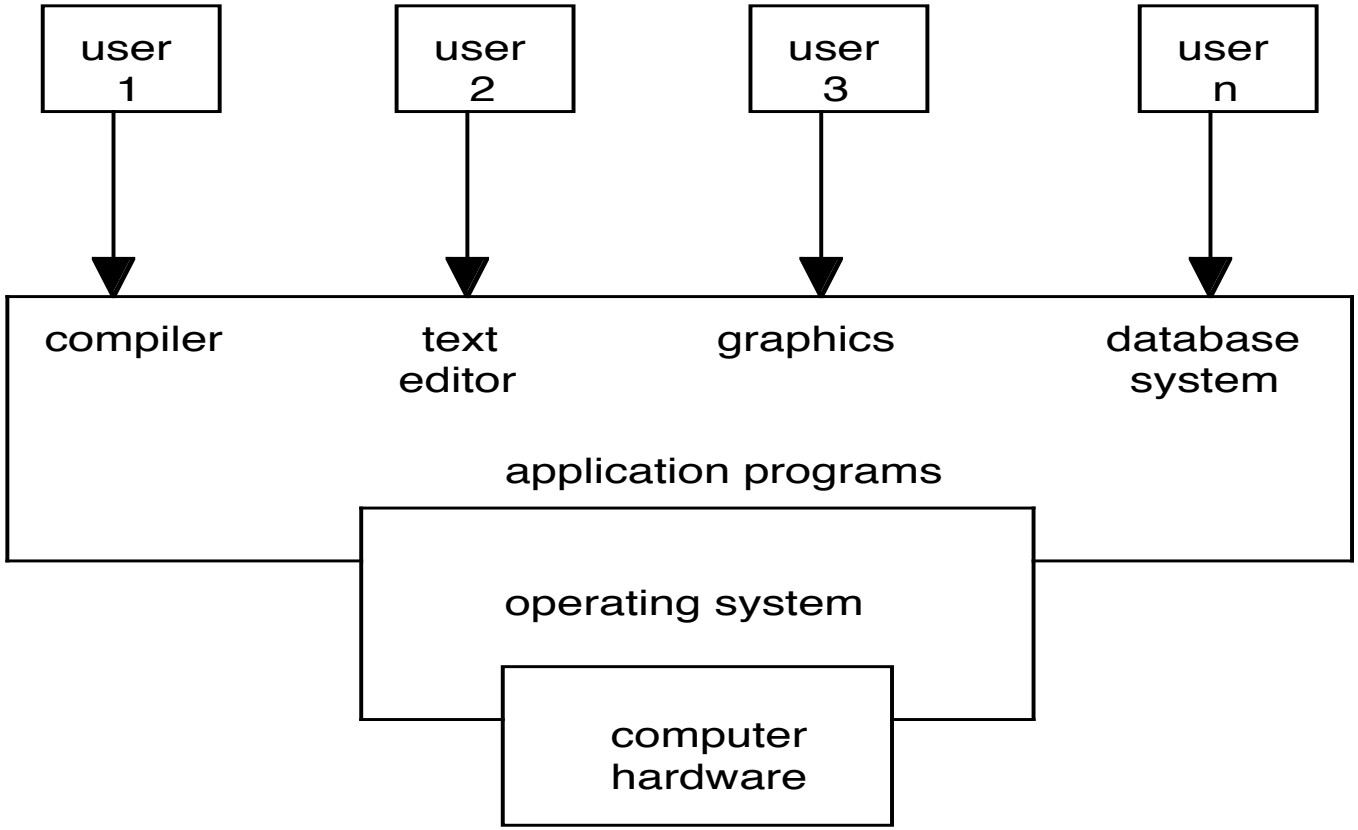
- From where to load a program into the memory in the past?
  - Punch cards, punch tapes, hard disks
- How to load a program in the early days?
  - A loading program is written to read cards/tapes and then store the instructions into the memory
  - One of the first device drivers
- Every programmer had to write the loading program!

Operating System

- Why not share the same loading program among the programmers?
  - The initial motivation for writing a piece of OS
- What is OS?
  - Includes kernel and many system programs
  - One or more programs which can be reused to conveniently use/share computer resources such as I/O devices
  - Control program
  - Wrapper
  - Resource allocator
  - Lazy government
  - Service provider

Computer Systems

- Abstract view of computer systems



OS concepts

- OS kernel
  - Provide the most basic and generic functionality
  - Handle resource sharing and I/O devices
  - Can only be used by system calls
- System calls
  - Used to get services from OS
- Routines/kernel functions
  - For system calls and interrupts

OS concepts (cont.)

- Daemons/servers
  - Achieves the functions of OS but running as a user program providing a service
  - Keep the OS kernel as small as possible
  - E.g. **sshd, syslogd, named, crond**
- Command shell
  - A user interface based on command line
  - A layer of software which wraps the OS kernel in more acceptable clothes
  - Allows users to interact with the machine, e.g. manipulate files, and run programs
  - Interpret scripts, e.g. Bourne shell, Bourne Again SHell (BASH), C shell, Python, Perl...

Shell programming (scripting)

- Why shell programming?
  - It is essential for system administrators (SAs)
  - Many administrative programs are written in scripts
  - The syntax is simple and easy to learn and understand
  - Quickly prototype a complex application
  - When efficiency and performance are essential, don't use shell scripts
- BASH(Bourne Again Shell)
  - Extended from the classic Bourne shell and Korn shell
  - A de facto standard for scripting on UNIX

Processes

- Process
  - A program which is being executed
  - Consists of program and its running state
- Why processes?
  - For convenient resource sharing
  - Time sharing: processes take turns to be executed
- Tasks/jobs/threads
- Scheduling of processes
  - Round-robin/queueing
  - Preemptive/non-preemptive

Processes (cont.)

- Context switching
  - Restore the state of a process when it is scheduled to run
  - The state of each process is described by a data structure called *process control block* (PCB)
- Inter-process communication
  - Pipes: enables one process to open another process as if it were a file for writing or reading, e.g. **ls -l | more**
  - Sockets: IP domain or Unix domain
  - Data sharing and synchronisation

Processes (cont.)

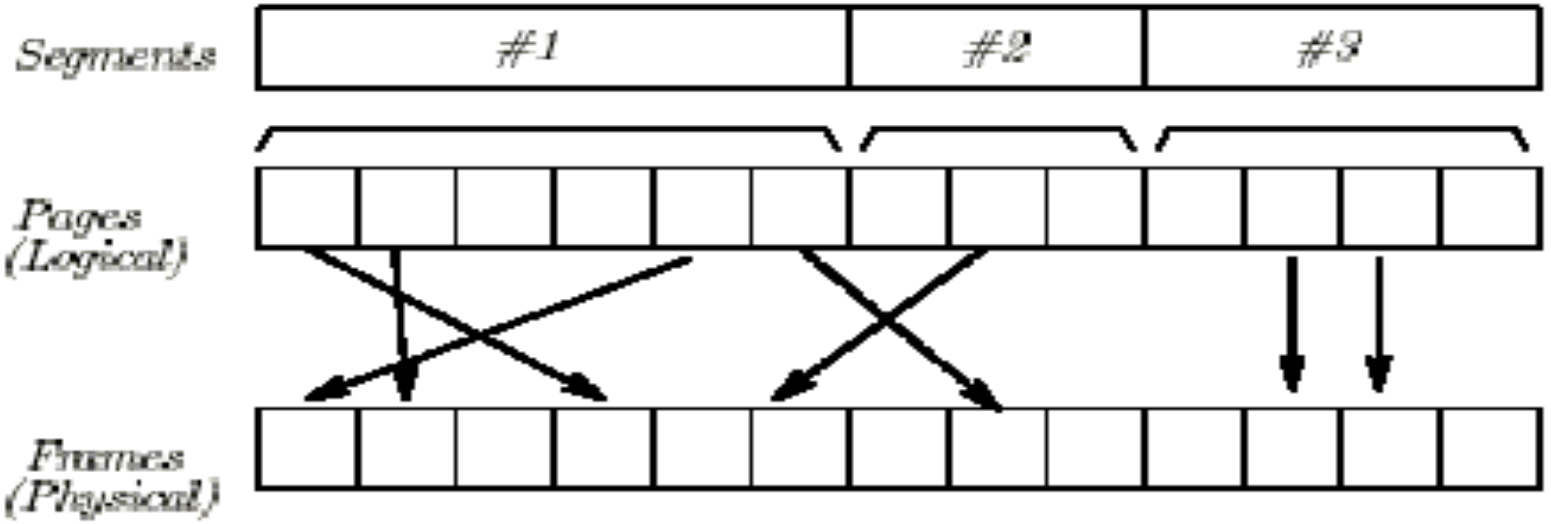
- Process creation
  - fork(), wait(), and exec()
  - Parent and child processes
- Process scheduling state
  - Ready: in line to be executed
  - Running: active
  - Waiting: sleeping or suspended
  - Terminated: defunct/zombie
- Deadlock
  - Due to resource sharing and waiting among processes

Memory system

- RAM/ROM
- Paging
  - Divided into pages, e.g. 4096 (4k) bytes
  - Addressing: page_id + offset
- Logical memory
  - To protect each program's memory space, logical memory space is used for each program
  - Each program has its own separate logical memory space starting at address zero, divided into code and data segment

Memory system (cont.)

- Logical vs. Physical memory
  - There is a mapping mechanism from logical to physical memory space. It is called a *page table*. It is loaded as part of the context switching
  - **malloc()** and **free()** to allocate and free logical memory.



Memory system (cont.)

- Virtual memory
  - A way of making the physical memory of a computer effectively larger than it really is.
  - Disk space (swap space) is used to store memory pages
  - Swapping: an entire process, including code and data, is expunged from memory
  - Paging: only some single pages are swapped out
- Page faults
  - Occurs when a page is not in the memory
  - When a page fault happens, the involved page has to be brought in from disk space.

Memory system (cont.)

- Paging strategies
  - When a page fault occurs, some page in memory has to be shifted out
  - FIFO
  - LRU
- Thrashing
  - a state where there are so many processes competing for limited resources that it spends more time servicing page faults and swapping in and out processes than it does executing the processes

I/O system

- I/O devices are handled with drivers
  - Drivers provide functions of device operations under a standard interface (a file) to user programs, e.g., open(), read(), and write().
  - Devices are operated by reading from/writing into I/O memory or ports
  - Drivers may be loaded as modules in Linux
- Character and block devices
  - Depend on if data are manipulated as a stream of bytes or blocks. Block buffer are used for block devices
- Synchronous and Asynchronous I/O

I/O system (cont.)

- Interrupts are used for devices with asynchronous I/O
  - A small piece of code in OS is used to handle each interrupt
  - Interrupt vector: contains addresses of the interrupt routines
  - Direct Memory Access (DMA): a device which copies blocks of data at a time from one place to the other, without the intervention of the CPU.
    - Very high speed devices could place heavy demands on the CPU if they relied on the CPU to copy data word by word.

File system

- File system
  - To directly operate disks (block devices) is troublesome and complicated
  - A file is a good abstraction for data stored on block devices
  - A file system is a high level interface to block devices such as disks. It consists of methods and data structures that an operating system uses to keep track of files on a disk
  - There are many different file systems, e.g. Linux supports ext2, fat, nfs, iso9660, …

Linux OS

- A member of the UNIX family
  - Inspired by MINIX
  - Initially developed by Linus Torvalds, but contributed by hundreds of developers around the world
  - Linux source code is under GNU General Public License, which is open to anyone to study
- Linux OS kernel
  - A true UNIX kernel, but not a full UNIX OS, as it doesn't include all applications such as windowing systems, compilers, and text editors which are available under the GNU license

Linux OS (cont.)

- Characteristics of Linux OS
  - A monolithic kernel
  - Supports kernel modules
  - Multithreading
  - Preemptive for 2.6 (but non-preemptive for earlier versions)
  - Multiprocessor support
  - Reconfigurable kernel
- Linux version
  - X.Y.ZZ - X.Y identifies the version number, ZZ is the release number. If Y is an even number, the version is a stable version.