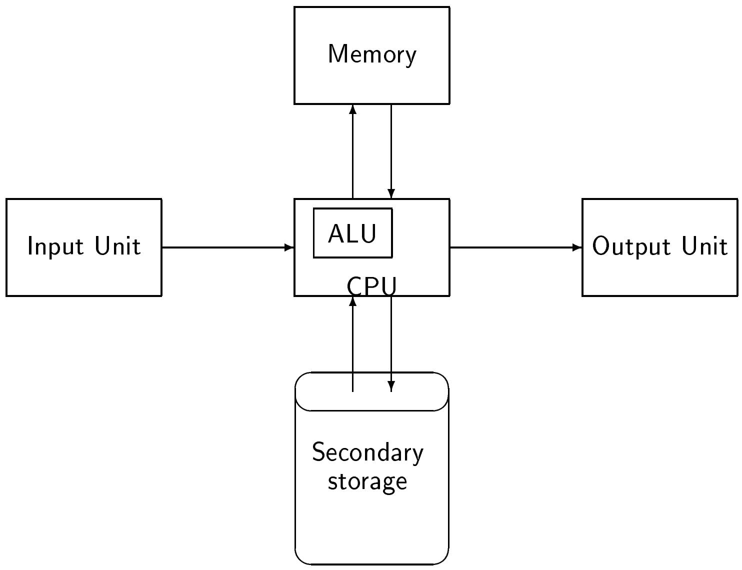
Introduction

Learning the lingo

- Computer
  - Collection of components that can perform arithmetic and logical calculations at incredible speed
- Hardware
  - Actual circuits and associated machinery that is visible as a computer
  - Keyboard, monitor, disk, tape, floppy drive, camera, microphone
- Software
  - Sequence of instructions/commands to make the computer perform a desired task
  - C is a language understood by the computer
- Program
  - A sequence of instructions in a language understood by the computer
  - Programs process data

Computer organization

- Loosely defined as in the following figure:



  - Input unit
    * Eyes and ears of the computer
    * Used to receive information from the outside world
    * Connects to different input devices
    * Keyboard, mouse, video camera
  - Output unit

    * Tongue/mouth, hands, and legs of the computer
    * Used to communicate information, or to do something
    * Monitor screen, printer, robotic arm
  - CPU or Central Processing Unit
    * Brain of the computer
    * Talks to every other unit via the nervous system (buses or wires)
    * Responsible for the behavior of every other unit in the system
  - ALU or Arithmetic and Logical Unit
    * Specialized part of the brain that is good at math and at making very simple decisions
    * May or may not be a separate unit compared to CPU
    * Generally referred to as a coprocessor
  - Memory
    * Used by the CPU to keep instructions and data
    * CPU picks up an instruction from memory and acts on it
    * Modified by the actions of the CPU
    * Also known as RAM or Random Access Memory
  - Secondary storage unit
    * Long term storage for programs that are not in active use at the time
    * Also used to keep data that needs to be stored for long term, such as files in an office

Batch processing, multiprogramming, and timesharing

- Batch processing
  - Job or task – name for a certain process, may consist of running more than one program
  - Processing jobs one after the other
  - Used in older computers, has different connotations now
- Multiprogramming
  - Keeping many programs in the memory at the same time
  - The CPU may divide its time between these jobs
- Timesharing
  - Requires interactive computing over terminals
  - Time is evenly divided amongst users sitting on different terminals

Personal computing, distributed computing, and client/server computing

- Personal computing
  - Started with the introduction of Apple computers in 1977
  - IBM created a standard for the PC in 1981
  - PCs are based on the microprocessor by Intel and almost invariably run the operating system from Microsoft
  - Some PCs run the OS/2 system from IBM while the free Unix for PCs – Linux – is fast gaining in popularity with the people who demand power

- Distributed computing
  - Early PCs could be linked together in computer networks
  - Today's computers are powerful enough to perform multiple tasks and to handle several users simultaneously
  - The idea is to distribute the load across several PCs rather than a single mainframe computer
- Client/server computing
  - Best exemplified by Netscape
  - Information on the WWW is kept on the WWW server and is transmitted to the Netscape or Explorer client when requested
  - The information is formatted at the client site
  - Other examples of client/server computing include the X window system

Different number systems: binary, hexadecimal, and decimal

Machine languages, assembly languages, and high-level languages

The Unix operating system

- Operating system
  - A collection of programs that control and allocate resources in a computer system – both in hardware and software
- Unix is multi-user, multi-tasking operating system developed at the AT&T Bell Labs
- Mostly developed through collective efforts by hackers
- It can perform functions like
  - Text/code editing and printing
  - Document handling and storage
  - Programming and software development
  - E-mail, fax, networking, WWW
- Versions of Unix
  - AT&T SYS V
  - BSD
- Student environment for Unix at UMSL
  - Unix and X windows in the SSB lab
  - Terminal-based Unix accessible from other computers and over modems
  - Complete Unix and X windows on the home computer, via Linux (most recommended)
- Logging in
  - Type in your username and password
  - The system sets up a working environment for you, controlled by the *hidden* files called `.cshrc` and `.login`
  - If you get a "login incorrect" message, you need to retype your username and password

  - Unix is case sensitive, and therefore, you should be careful in what case you type (uppercase letters v. lowercase letters)
  - A general convention is to work in lowercase as much as possible, keeping the uppercase free for special things

- Logging into other machines
  - `telnet`
  - `rlogin`
- Logging out
  - Use the `logout` command
  - The `exit` command
  - The `^D` command (end of file)
  - Using the mouse to kill a window
  - Using the mouse to conclude a session
- The Unix shell
  - Provides the environment to work in
    * Interprets the commands and executes them
  - Allows you to *talk* to the computer
  - Different shells include the Bourne shell, the Korn shell, the C shell, `bash`, `tcsh`
  - On campus machines, you have access to the C shell, the Bourne shell, and `tcsh`
  - Your default shell is the C shell
- The shell prompt
  - Shell's way of saying that it is ready and waiting for your commands
  - For illustration, we'll use % to indicate the C shell prompt and $ to indicate the Bourne shell prompt (also the defaults for these shells)
  - The prompt can be customized by using the command `set prompt`
    * The following command sets the prompt to `Yes Master`
      ```
      % set prompt = "Yes Master "
      Yes Master
      ```
- Entering a command
  - Command is nothing but a program that resides on the disk
  - Commands are case sensitive, and generally consist of a single word
  - Checking the date and time
    ```
    % date
    Thu Aug 29 09:50:24 CDT 1996
    ```
  - Checking who all are logged in on the machine
    ```
    % who
    ```
  - Checking your own identity
    ```
    % who am i
    ```
  - Finding the identity of others
    ```
    % finger
    ```
- Correcting mistakes

  - Use the backspace key
  - If it does not work, you can force it to work by using the command
    `% stty erase <backspace>`
  - Erasing the whole line
    `^U`

- Syntax of Unix commands
  - Unix commands are entered as

    command [option(s)] <parameter(s)>

  - Almost all commands are in lowercase
  - Options are like switches in DOS and modify the behavior of the command
  - All options are prefixed by a dash
  - Parameters are operated upon by the command, they form the data for the command
  - Command, options, and parameters are separated by space character
  - You can write two or more commands on the same line if you separate them by a semicolon (;)
  - You can extend the command to a second line by typing a backslash \ follwoed by the return key

- The text editor `vi`
  - Stands for Visual
  - Started by typing the command

    `vi <filename>`

  - Two modes – command and input

History of C

- BCPL, 1967
  - Designed by Martin Richards in 1967
  - Useful for writing system software like compilers and operating systems
- B
  - Developed by Ken Thompson for early versions of Unix
  - Inspired by BCPL
  - Typeless like BCPL
- C
  - Developed by Dennis Richie in 1972
  - Modified from B by adding type support and other features
  - Meant to be portable
- ANSI standard for C
- Portability myth

C standard library

- Collection of functions
- Building block approach
  - If a function is not available, write it yourself
- Software reusability
- preferable to use ANSI standard functions
- Structured programming

C programming environment

- Editor, source program
- Preprocessor/Compiler
- Linker/Loader
- Executable version
- All files have an extension `.c`
- The KISS principle
- Your best teacher