OS Examples

- Unix
- Minix
- Linux
- TinyOS
- Palm OS
- Symbian OS

UNIX History

- MULTICS (MULTiplexed Information and Computing Service) – Bell Labs
- UNICS (Ken Thompson): Uniplexed Information and Computing Service -> UNIX
- PDP-11
  - Large memory (for its time)
  - Memory protection h/w
- Language
  - Originally written in assembly language, but difficult to port to new machine
  - Needed high-level language, B (new) (Thompson)
  - Rewritten in C (Denis Ritchie)
- Users
  - Universities -> industry
  - Minicomputers, workstations

UNIX Versions

- Porting to a new machine
  - Write a C compiler for new machine
  - Write device drivers for new machine
  - Write machine-dependent code (interrupts, memory management)
  - => portable C compiler (Steve Johnson)
- AT&T Unix (Bell's parent)
- Berkeley UNIX
  - Berkeley Software Distribution (BSD)
    - Virtual memory, lon file names, networking (TCP/IP)
- Standard UNIX
  - POSIX (Portable Operating System Unix)
  - Intersection of above
- Open Software Foundation (OSF)
  - IBM, DEC, Hewlett-Packard

Unix Overview

- Goals
  - Interactive timesharing system
  - Designed by programmers for programmers
  - Facilitate complex s/w development projects
  - Simple, elegant consistent
    - principle of least surprise
  - Power and flexibility
    - Small number of basic elements can be combined in an infinite variety of ways
    - Each program should do just one thing
  - Servant not a nanny

UNIX Features

- Unix has several layers, like a pyramid
  - h/w: CPU, memory, disks etc
  - UNIX OS: process management, memory management, file system etc (kernel mode)
  - Standard library: open, close, read, write etc
  - Standard utility programs: shell, editors, compilers etc
  - Users
- Login for each user
  - Password file with one entry for each user
- Shell
  - Commands, flags, standard input/output/error, redirection, filters, wildcards, pipe symbol
  - Example: sort < in | head -20

UNIX Features

- Files and directories
  - No distinction between ASCII, binary or other types of files
  - File owner determines meaning of bits in file
  - Names: 255 chars
  - Rights bits: 9-bit mode for each file rwxrwxrwx
    - First 3 bits refer to the owner's access
    - Next 3 bits to other members of the owner's group
    - Last 3 bits to everyone else
  - Directories with / as component separator
  - Standard directories: bin, lib, usr, tmp, etc

Unix Features

- Unix provides many utility programs
  - File and directory manipulation commands
    - cp (copy), mv (move)
  - Filters
  - Compilers and program development tools
    - Make: used to maintain large programs whose source code consists of multiple files
  - Text processing
  - System administration
  - miscellaneous

Unix Concepts

- Processes
  - Each process runs a single programme
  - Multiprogramming system => multiple independent processes running at the same time
  - Background processes, **`daemons`**, are started automatically when the system is booted
    - `cron daemon` wakes up every minute to check if there is any work for it to do
  - **Parent** process can **fork** a **child** process
    - Each have their own private images
    - Open files are shared between parent and child
  - Processes in Unix can communicate with each other
    - using a form of message passing via channels called **pipes**
    - Via software interrupts

Unix Concepts

- Memory model
  - Quite straightforward (for portability reasons)
  - Each process has a text (code), data and stack segment
- File System
  - File descriptor (when opening file)
  - Absolute and relative path
  - Working directory
  - Link: a new directory entry that points to an existing file
  - Separate file systems can be mounted so the user dot have to know where each file (physically) resides
  - Locking: can lock bytes or files
    - Shared and exclusive locks

Unix Concepts

- Input/Output
  - **special files** allow access to devices
  - Each I/O device is assigned a path name, usually in /dev
    - e.g. the printer might be /dev/lp
    - cp file /dev/lp (to print a file)
    - No special mechanism need for doing I/O
    - Usual file protection rules apply to I/O devices
- System calls
  - Every system call returns a status code telling whether the call was successful or not (and an error number in a global variable *errno* if an error occurred)

Unix Concepts

- System calls
  - Process management
    - FORK: the only way to create a new process
    - EXEC: ~ to run system commands
  - Memory management
    - Via `malloc` library procedure
  - Files and directories
    - `creat, read, write, chmod`
  - Input/output
    - Many different calls, usually with a return code

Unix Implementation

- Processes
  - User part and kernel part
  - Kernel keeps process table and user structure for each process
  - Timesharing system => provide good response to interactive processes
    - Low-level algorithm picks the process to run next from the set of processes in memory and ready to run (uses multiple queues with different priorities)
    - High-level algorithm moves processes between memory and disk so that all processes get a change to be in memory and run

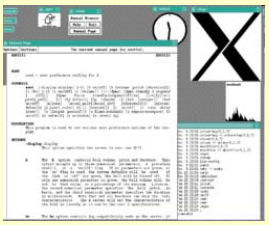
Unix Implementation

- Memory management
  - Originally swapping, but later paging
  - Page daemon
- File system
  - 3 tables: file descriptor table, open file description table, i-node table (with accounting information)
- Input/Output
  - Block device I/O uses a buffer cache (with LRU algorithm)
  - Character I/O
    - Raw – unchanged
    - Cooked – raw data is processed

Unix: Other

- Year 2038 Problem
  - Time from Jan 1, 1970
  - Solution in 64-bit apps (292 billion years)
- Unix/UNIX/UNIX
- Unix-like
  - Legal issues

X Window System 1990s



Minix

- Minimal Unix
- Kernel type: microkernel
- License: free and open source s/w (permissive free software license)
- Developer: 1987 Andrew Tanenbaum (Vrije Universiteit in Amsterdam)
- Usage: mainly for educational purposes
  - Teaches how kernels should really work
- Inspired the creation of Linux
- Current version: Minix 3
  - Industrial ripe kernel for embedded applications

Linux

- OS Family: Unix-like
- Latest (unstable) release: 15/04/09)
- Kernel type: monolithic
- License: various free, GNU General Public License (GPL)
- Model: free and open software collaboration
- Usage: mainly servers, embedded devices to supercomputers
  - Some governments: Brazil, India, China
- Linux:
  - Linux kernel - 1991 Linus Torvalds
  - Rest of system (including utilities and libraries) GNU OS 1983 (Richard stallman)

Linux

- Design
  - Unix-like OS
  - Unix principles from 1970s and 1980s
  - Monolithic kernel handles
    - Process control, networking and peripheral and file systems access
  - Device drivers are integrated directly with the kernel
  - GNU userland
    - C library, shell, many common Unix tools
  - GUI based on X Window System
- User Interface (UI)
  - command line interface (CLI) or GUI
  - “Headless system” (no monitor, remote control)

Linux

- Development
  - **Copyleft**: any work derived from a copyleft piece of s/w must also be copyleft itself
  - Aim for interoperability with other OSs
  - Try to adhere to POSIX (only Linux-FT certified)
  - Collaborative yet independent development
  - **Linux distribution** (distro) is a project that manages a remote collection of system s/w and application s/w packages available for download and installation through a network connection
    - Can adapt OS to individual needs
    - Mechanism for handling upgrades (e.g. Synaptic Package Manager)

Linux

- Community
  - Driven by developer and user communities
  - Debian (volunteer basis), Red Hat/Fedora (commercial)
  - Linux Users Groups (LUGs)
  - Corporations:
    - Large: Dell, IBM, Oracle, Sun, Novell, Nokia
    - Red Hat
  - Business model
    - Symbiotic
    - Charge for support
    - Specialised business version
    - Give away s/w to sell h/w

Linux

- Programming
  - Many languages supported
  - Graphical applications: GNOME, KDE
- Uses
  - Embedded, desktop
  - Servers (Linux, Apache, MySQL, Perl/PHP/Python – LAMP)
  - Supercomputers (88% of top 500 systems)
  - Widely ported
- Desktop Linux
  - Improving
  - Wine project: provides a Windows compatibility layer to run unmodified Windows applications on Linux
  - Lags behind in the field of gaming
  - Localised distributions

Linux

- Uptake
  - 12% of server market
  - 1-2% (?) of desktop
    - Frictional cost of switching, lack of support
    - Security, reliability, low cost, freedom from vendor lock-in
  - XO Laptop: One Laptop Per Child
  - Film industry
    - Titanic
    - 95% of animation companies use Linux
- Common distributions
  - Debian GNU/Linux, Fedora, Gentoo Linux, Mandriva Linux, OpenSUSE, Slackware, Ubuntu

TinyOS

- Free and open source component-based OS
- Target area: Wireless Sensor Networks (WSNs)
  - A wireless network consisting of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions
  - e.g. temperature, sournd, vibration, pressure
- Embedded OS
  - written in nesC, a dialeect of C optimised for the memory limitations of sensor networks
- Developer: TinyOS Alliance
  - Originally University of California, Berkeley with Intel
  - Now larger consortium
- Releases
  - Initial: 200
  - Latest: 2008

TinyOS: Features

- Built out of s/w components
- Components connected using interfaces
- Components include:
  - Packet communication, routing, sensing
- Completely non-blocking
  - Has a single stack
  - Asynchronous I/O with callback
  - Programmers must manage many small event handlers
  - Has non-preemtive task (similar to interrupts)
    - FIFO order
- Good for I/O, not so for CPU-heavy applications

Palm OS

- Proprietary, embedded OS to power mobile devices
  - Smartphones, wrist watches, handheld gaming consoles, barcode readers, GPS devices
- Designed for ease of use with a touchscreen-based GUI
- Suite of basic applications for personal information management
- Origin:
  - Palm computing for personal digital assistants (PDSs) in 1990s
  - Various rights bought since then

Palm OS: Features

- Simple, single-tasking environment to allow launching of full screen apps with a basic, common GUI
- Simple security model
  - Device locked by password
  - application records can be made private
- TCP/IP network access
- Other:
  - Handwriting recognition system (Graffiti 2)
  - HotSync technology for synchronisation with desktop computers
  - Sound playback
- Many lawsuits over the years

Palm OS

Symbian OS

- Proprietary
  - Ericsson, Nokia, Motorola, Psion
  - Now Nokia
- Embedded OS for mobile devices
  - Libraries, UI, tools
- Kernel: microkernel
- Pre-emptive multitasking, memory protection
- 3 design rules:
  - Integrity and security of user data
  - User time must not be wasted
  - All resources are scarce

Symbian OS

- Request-and-callback approach to services
- Separation between UI and other apps
- Optimised for
  - low-power battery-based devices
  - ROM-based systems
- OS kernel supports good real-time response
  - Can build a single-core phone (i.e. one in which a single processor core executes both the user apps and the signalling stack)
  - Not available in Linux
  - Smaller, cheaper, more power efficient phones

Symbian OS

- Layers
  - UI, appls, OS Services, Base services, kernel services and h/w interface
- Market share (Nov 2008):
  - Symbian OS 47%, Apple (iPhone OS) 17%, MS (Windows CE and Windows Mobile) 13%
- Not yet open source
  - Phone manufacturers and other partners have access to parts of source code
  - API are publicly documented
  - Plans exist to make it open source
- Security
  - Subject to a variety of viruses (e.g. Cabir)
  - Improved security features