The Linux Operating System

Presented by Sebastian Smith

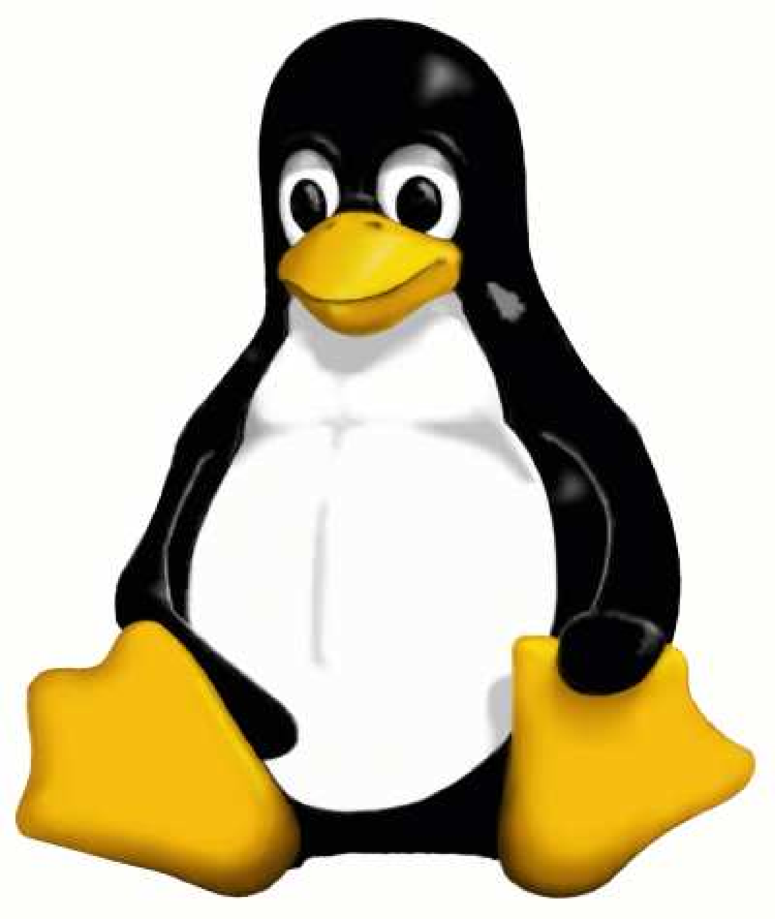
CS 446: Principles of Operating Systems

November 1, 2005

Overview

- Introduction
- GNU/Linux
- Distributions
- Distribution Case Studies
- Questions

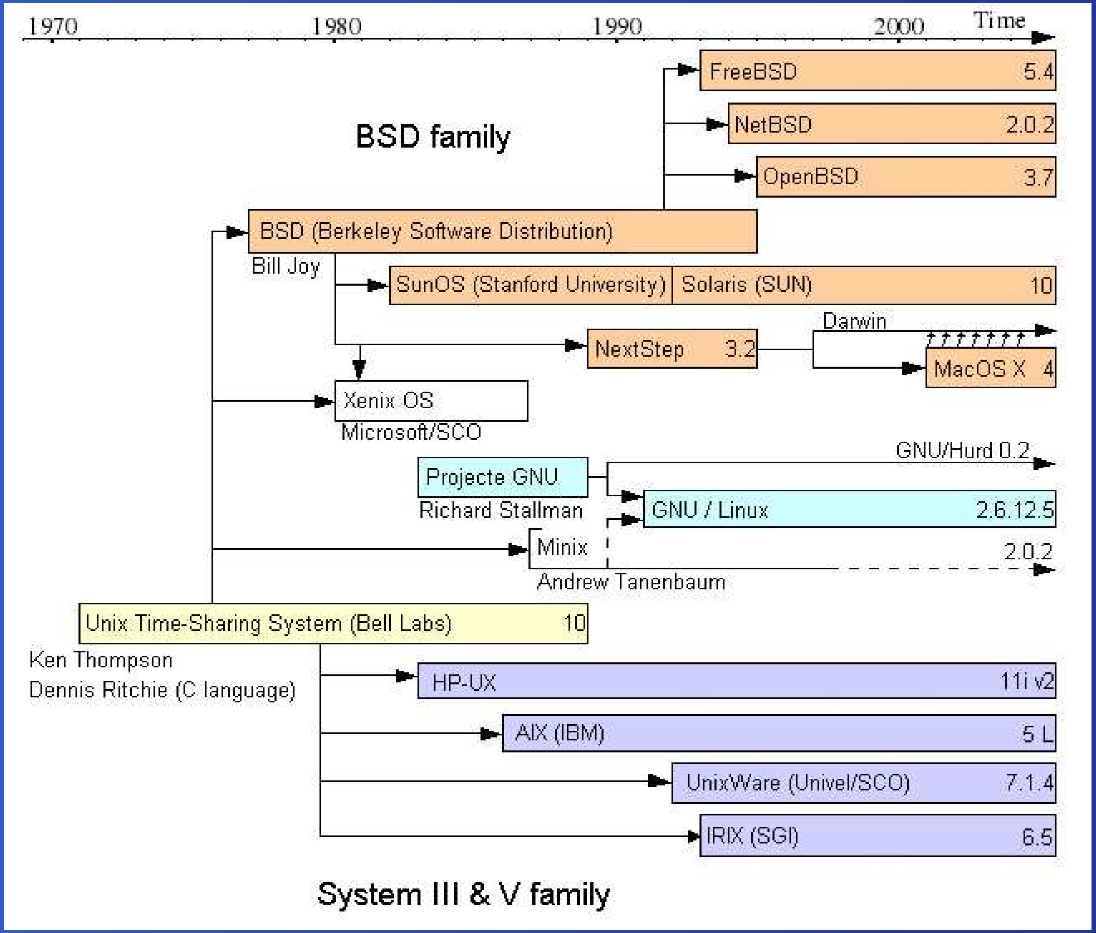
What is Linux?



What Linux Is

- A kernel
- Originally written by Linus Torvalds
- Released in 1991 (Windows 3.0a)
- Originally written to run on the Intel 80386
- Now runs on a variety of architectures

The History of *nix



GNU/Linux

- The Linux operating system
  - The Linux kernel
  - GNU libraries and tools
- Linux would not exist without GNU

Linux Distributions

- The Linux kernel packaged with operating system and other software
- Released by companies, communities, and individuals
- Quality control: software packages are assembled and tested before distribution
- Designed for specific audiences
- Currently 386 Linux distributions

Common Distributions

- Ubuntu
- Mandriva
- SUSE
- Fedora
- Slackware
- MEPIS
- KNOPPIX
- Debian
- Damn Small
- Gentoo

Ubuntu: Linux for Human Beings

- Ubuntu = "humanity to others"
- Free of charge
- Based upon Debian Sid
- Easy to install
- Apt package management (DEB)
- LiveCD

Gentoo

- Source based distribution
- Package management based on BSD Ports
- Highly customizable
- Highly optimized
- Excellent community
- Targeted at advanced users

Review

- GNU/Linux
- Distributions
- Distribution Case Studies
- Questions

Questions?

The Linux Kernel

Presented by Sebastian Smith

CS 446: Principles of Operating Systems

November 1, 2005

Overview

- Introduction
- Processes and threads
- Memory Management
- CPU Scheduling
- The Virtual File System
- Questions

The Linux Kernel

- Monolithic kernel
- Loadable modules (microkernel-like)
- Drivers can run in ring 0 or in userspace (ring 3 in x86)
- 10,239 lines of code at version 0.01
- 5,929,913 lines of code at version 2.6.0
- Current stable release 2.6.14

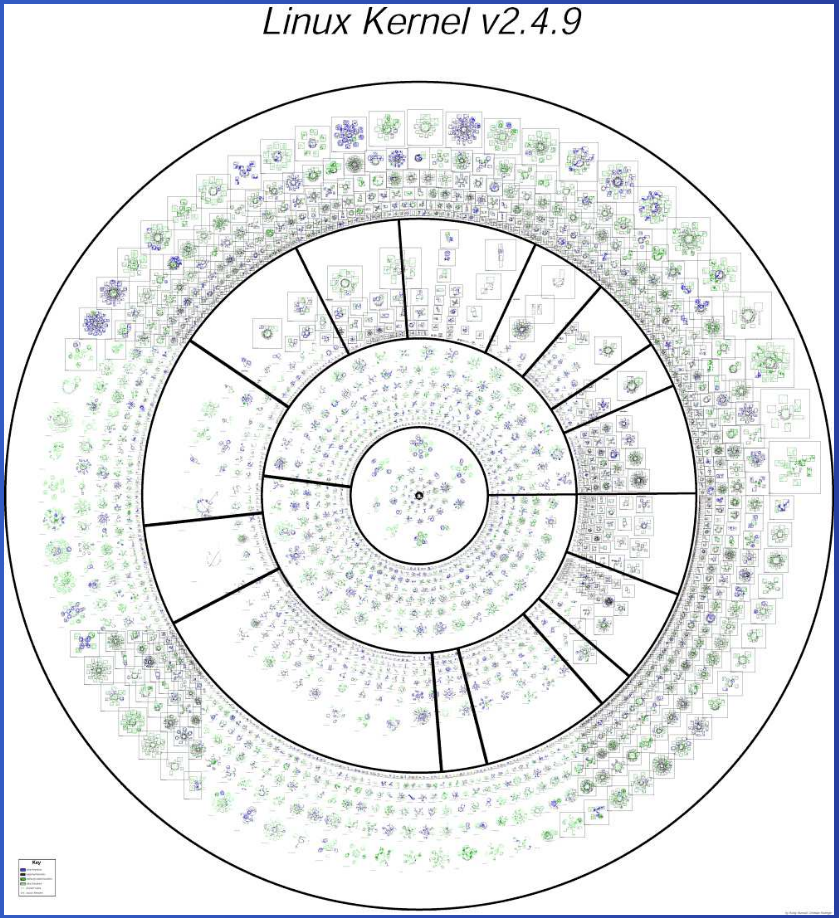
Version Numbering

- Three number version scheme A.B.C[.D]
- A denotes the kernel version
- B denotes the major revision (odd = development version)
- C denotes the minor revision
- D optionally denotes the fix of a grave error

Kernel Component Examples

- Processes and scheduler
- File systems
- Virtual memory
- Network protocols
- Device drivers
- Signal handling

Kernel Diagram



Processes and Threads

- Supports multiple executable file formats including ELF and a.out
- Processes implemented as a vector of tasks
- Number of processes limited by size of task vector (512 by default)
- 2.6 kernel support up to one billion processes, 2.4 up to 32 thousand

Processes and Threads (Cont)

- No distinction between threads and processes ("lightweight processes")
- Multiple user-level threads are mapped into a single kernel-level process that share GID
- Process created by copying the attributes of the current process
- Sharing of virtual memory causes thread functionality

Memory Management

- Virtual Memory Addressing
  - Three level page table
  - Page directory
  - Page middle directory
  - Page table
  - Page allocation based on the buddy system
  - Page replacement based on the clock algorithm

Memory Management (Cont)

- Kernel Memory
  - Uses virtual memory page allocation mechanism
  - Buddy system used to allocate and deallocate memory
  - "Slab allocation" for odd sized memory allocation

CPU Scheduling

- Three Linux scheduling classes
  - SCHED_FIFO (real-time) [0–99]
  - SCHED_RR (real-time) [0–99]
  - SCHED_OTHER (non-real-time) [100–139]
- Scheduling priorities may be used within each class
- A lower priority number = higher priority

SCHED_FIFO

- Higher priority threads interrupt (Preemption)
- Blocking will interrupt
- Yield will interrupt
- Interrupted thread is put in a priority queue

SCHED_RR

- Similar to SCHED_FIFO
- Associates a timeslice with each thread
- Once time quantum expires the thread is placed at the end of its priority queue.
- *Programming tip*: Keep a shell with a higher priority open at all times to kill test applications.

SCHED_OTHER

- The default Linux scheduler
- Called the O(1) scheduler in the 2.6 kernel
- Selection of a process and assigning it to a processor is done in constant time
- Process assigned to the "active" priority queue when created
- Once timeslice is completed process is moved to "expired" priority queue

SCHED_OTHER (Cont)

- Once all processes have been run in active queue pointers are switched between active and expired queues
- Round robin is used to schedule processes within the active priority queue
- Favors I/O bound tasks over processor-bound tasks
- Allows for dynamic priorities [100 – 139]

The Virtual File System (VFS)

- Presents a single, unified file system interface to user processes
- Defines a common file model
- Assumes files are objects on local mass storage regardless of the target file system or underlying hardware
- Files within the VFS have properties
- A mapping module transforms the VFS representation to the real file system

Review

- Processes and threads
- Memory Management
- CPU Scheduling
- The Virtual File System
- Questions

Questions?