The Linux Operating System

Presented by Sebastian Smith

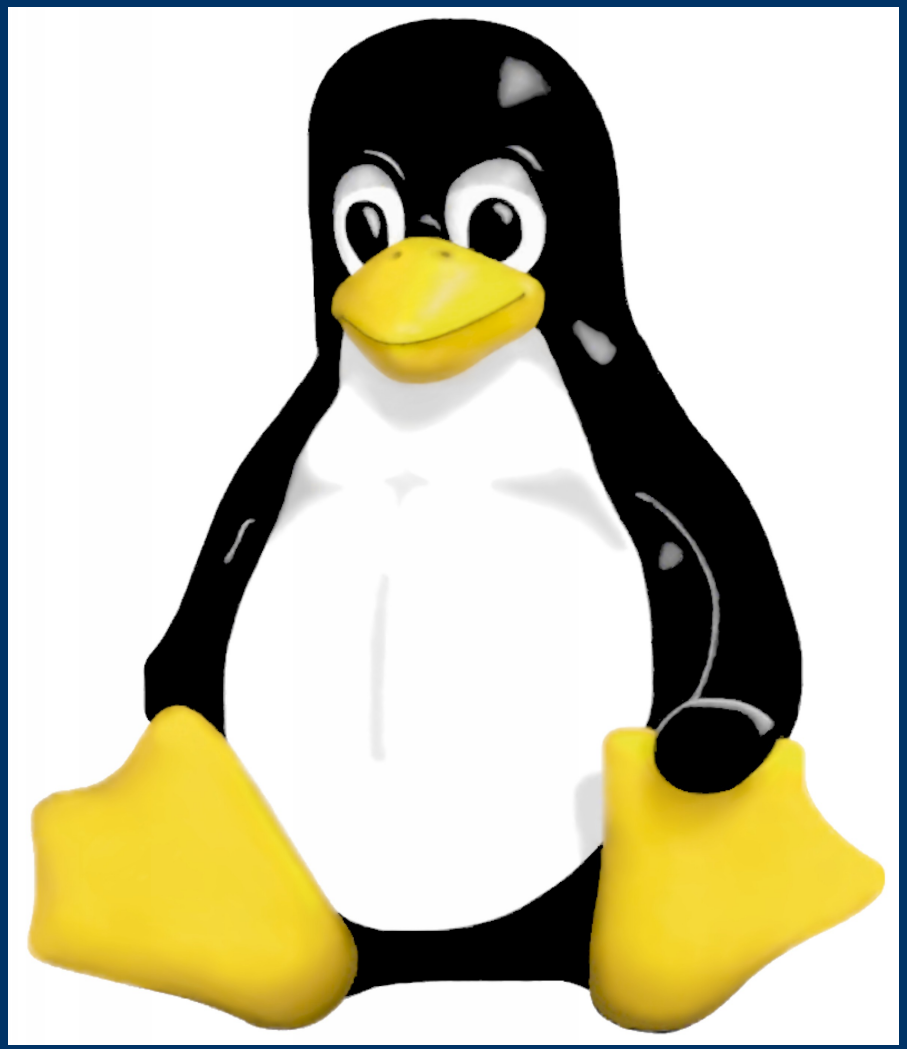
CS446: Principles of Operating Systems

4/4/06

Overview

- Introduction to Linux
- The Linux Kernel
- Processes
- Threads
- Memory Management
- CPU Scheduling
- The Virtual Filesystem

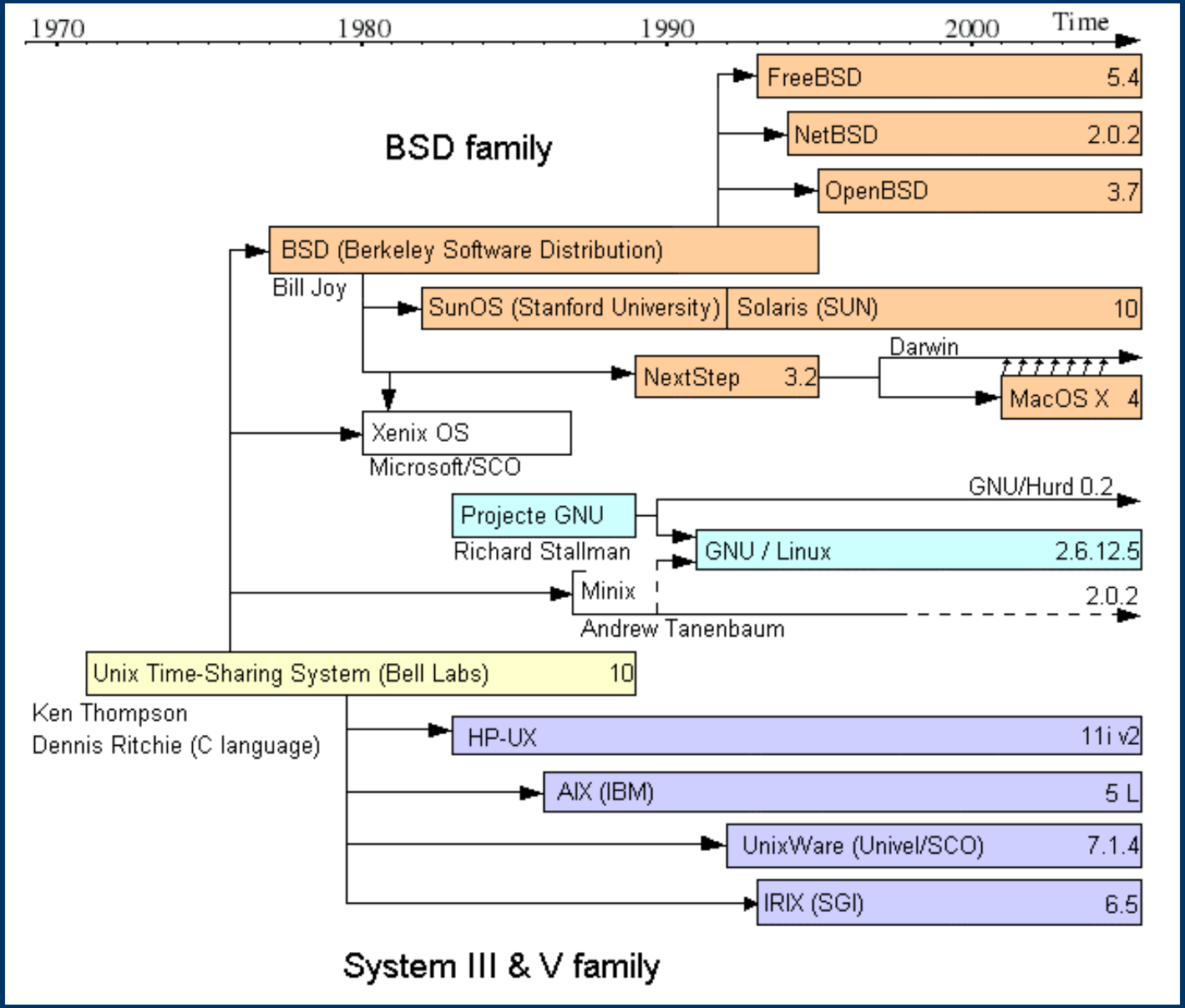
What is Linux?



What Linux is

- A kernel
- Originally written by Linus Torvalds
- Based on Minix by Andrew Tanenbaum
- Released in 1991 (Windows 3.0a)
- Originally written for the 80386
- Supported by a community of hackers

Linux in the Unix Family



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 software
- Released by companies, communities and individuals
- Quality control
  - Software packages are assembled and tested before distribution
- Designed for specific audiences
- Currently ~386 distributions
- Full installation, embedded, and virtual distributions

Common Distributions

- Ubuntu
- Mandriva
- Suse
- Fedora
- Slackware
- Mepis
- Knoppix
- Debian
- Damn Small
- Gentoo

Supported Architectures

- IA-32
- IA-64
- x86
- x86_64
- Alpha
- MIPS
- Motorola 68k
- Sparc
- Ultrasparc
- PPC64
- HP PA-RISC
- Cell
- ARM
- PPC
- Hitachi SuperH
- IBM S/390
- DEC VAX
- AXIS CRIS

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 2.6.16

The Linux Kernel (cont)

- 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

The Linux Kernel (cont)

- Primary Kernel Components
  - Process/thread and I/O schedulers
  - File systems
  - Virtual memory
  - Network protocols
  - Device drivers
  - Signal handling

Processes

- Multiple supported binary formats
- 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

Threads

- 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
- Thread functionality results from sharing of virtual memory
- Native POSIX Thread Library

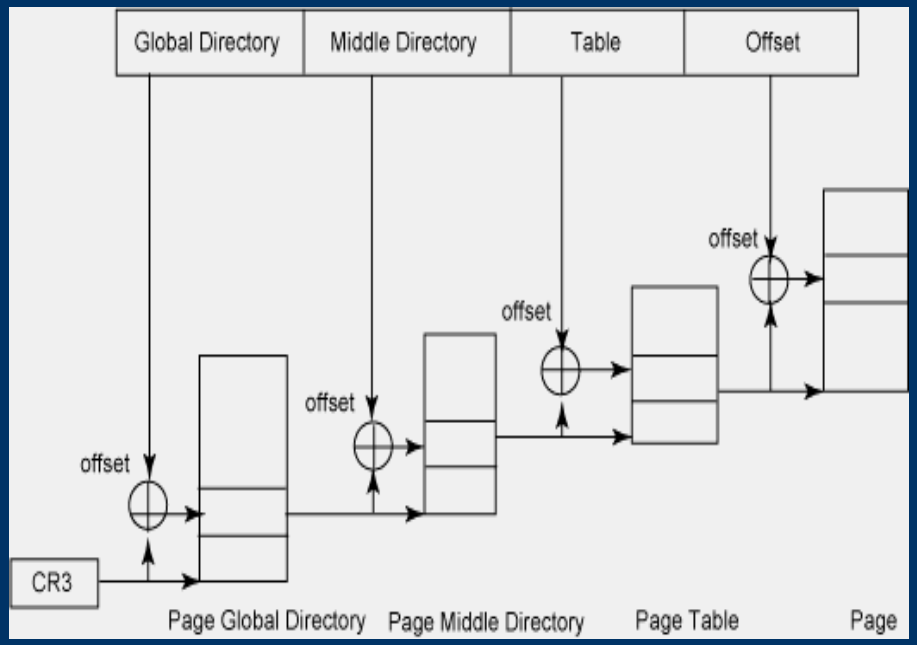
CPU Scheduling

- Three Linux scheduling classes
  - SCHED_FIFO (real-time) [0–99]
  - SCHED_RR (real-time) [0–99]
  - SCHED_OTHER (non-real-time) [100–139]
- Preemptive scheduling
- Dynamic priorities
- Scheduling priorities may set within each class
- A lower priority number = higher priority
- Switching to level [0-99] requires root access

CPU Scheduling (cont)

- The O(1) scheduler (SCHED_OTHER)
  - Scheduling done in constant time
  - Scales well with number of processes and processors
  - Improved SMP affinity
  - Provide good interactive performance
  - Favors I/O bound tasks over processor-bound tasks
  - Runqueue (list of runnable processes)
    - Active and expired priority arrays

Memory Management



- Virtual memory
  - Three level page table (x86)
  - 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

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
- A mapping module transforms the VFS representation to the real file system

Questions/Discussion