

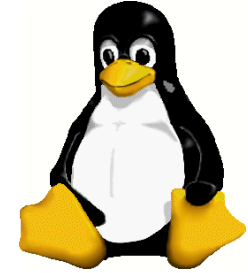
Linux

What is it ?

What's good about it ?

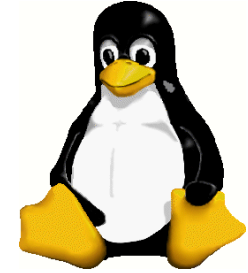
What's bad about it ?

History



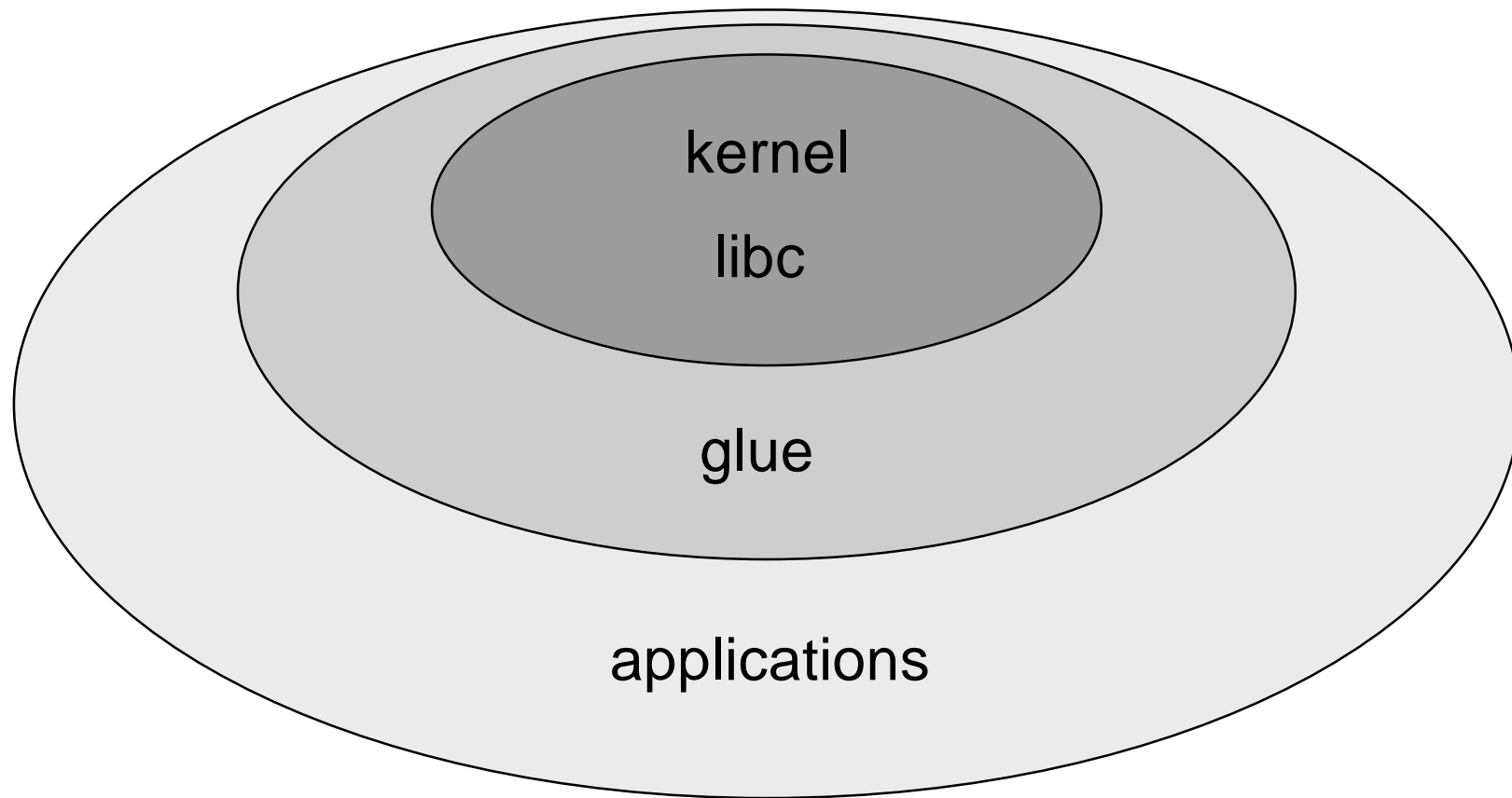
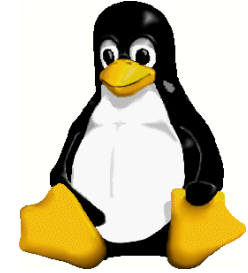
- Minix by Tanenbaum in late 1980s
- Linus Torvalds started Linux as a hobby project to improve on Minix
- First official public version late 1991
 - » kernel, libc and a few ported gnu tools
- Version 1.0 early 1994
 - » complete system, loads of ported software
- Now on version 2.0

Linux today

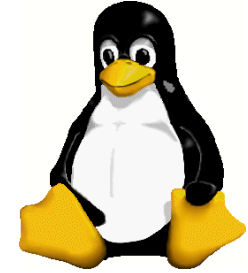


- Public domain, freely redistributable version of System V Unix
- close to POSIX / XPG4.2 compliance
- runs on Intel-based PCs, Alpha-based PCs, Suns, SGIs, Macs, HP, DecStations
- how many using it -- 2 - 3 million

What is Linux ?

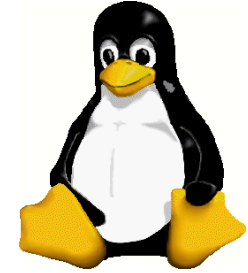


Who maintains it ?



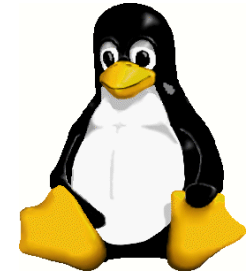
- Coordinated teams of volunteers
- widely geographically distributed
- eg Linus coordinates kernel
- libc (GNU libc team)
- ext2 filesystem (Stephen Tweedie et al)
- TCP/IP stack (Alan Cox et al)

Distributions



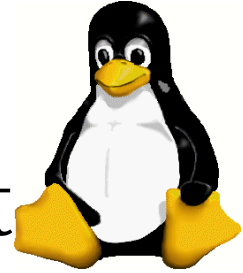
- DIY
 - » No support, much effort, most up-to-date ?
- Volunteer Distributions
 - » Free - no support
 - » eg. Slackware
- Commercial Distributions
 - » Free - no support
 - » Buy - support + some proprietary tools
 - » eg. Redhat, Caldera, Debian

Redhat



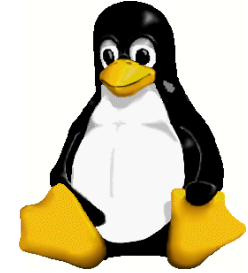
- No. 1 commercial distribution
- Most carefully put together sets of s/w
- Most architectures supported
- Most importantly - RPM

Redhat Package Management



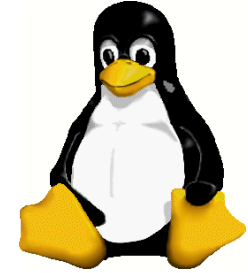
- Similar to Sun's *pkgadd*
- an RPM package is a
 - » container of files
 - » pre install script
 - » post install script
- RPM checks
 - » for filename clashes
 - » dependencies

Source RPMs



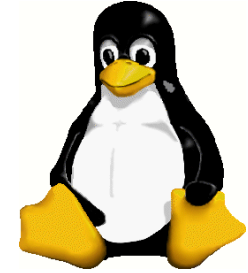
- RPM has source RPMs - container for sources
 - » original sources
 - » patches to work on Linux
 - » local patches
- RPM when building from source RPM
 - » unpacks original sources and applies patches
 - » builds and installs binaries
 - » builds a binary RPM for distribution

Why use RPMs ?



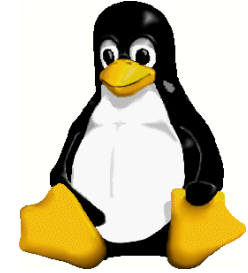
- ✓ RPMs available and maintained for much of the public-domain software we use
 - ➔ less effort to maintain PD software
- ✓ Easier to distribute RPMs to home and portable machines
- ✓ Easier to have different versions of software on different machines
- ✗ Wasteful when distributing small changes

Linux - security



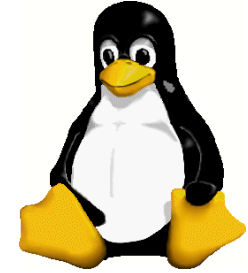
- Holes (as CERT report them...)
 - ✓ Redhat very responsive to CERT notices
 - ✓ have sources, so can fix yourself
- Capabilities
 - ✓ PAM (Pluggable Authentication Modules)
 - ✓ sources - easy to add / modify
 - ✗ *secure* NFS status unknown

Linux - performance



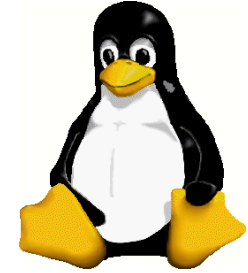
- ✓ System generally faster than Solaris
 - » Faster system calls (lighterweight)
 - » Memory requirements considerably less
- ✗ NFS performance is a concern
 - » but is improving
 - » no NFS V3 (yet...)
 - » other network performance as good as Solaris
- ✗ SMP performance is variable
 - » system calls suffer

Linux - reliability



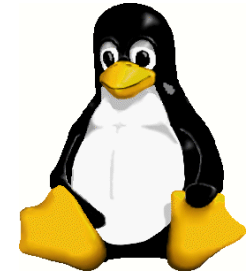
- No real concerns
 - » as long as don't use latest kernel version
- Intel PC and Alpha PC platforms most stable
- Sun platform stable on most Suns
- SGI and Mac platforms immature

Software



- ✓ Most of what we run today is available
- ✗ WWW browsing
 - » Netscape for Linux lags behind
- ✗ Java development
 - » Sun don't port JDK to Linux
 - » Volunteer port lags behind
- ✗ C / C++ development
 - » gnu compilers
 - » no decent debuggers, class browsers etc

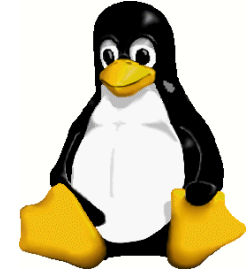
Support Effort



- ✓ Most public domain software already in RPM
- ✓ Can choose to have an evolving system
- ✓ Easier to integrate local system glue

- ✗ Some problems with RPM (eg small changes)
- ✗ Greater temptation for *versionitiss*
- ✗ Greater need to understand hardware

Where are we now ?



- Experience
 - » CS3 lab (since Dec 95)
 - » esk and tay
- Installation / configuration tools
 - » Network install using lcfg
 - » porting of lcfg scripts (not perl based yet)
- Nearly can support many machines with configurations following lcfg