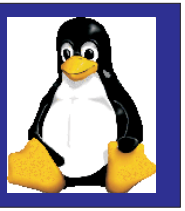
Linux for Mainframers



Laura Knapp
Technical Evangelist
1-919-224-2205
Laura@lauraknapp.com
www.lauraknapp.com

Agenda

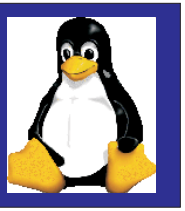
Background

Installing Linux

Major Components of Linux

Resources

History



Fastest growing operating system today

Unix-like operating system
Named after Linus Torvalds
Linus + Unix = Linux

Renowned for speed and reliability

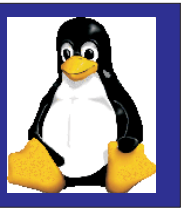
Runs on many different kinds of hardware

Why are there special releases for the mainframe?
- To support the mainframe devices
- Mainframe folks can't handle frequent releases

Several vendors sell very affordable "distributions"

And it's FREE to download

Continued Innovation



Linus Torvalds originated

Open source code

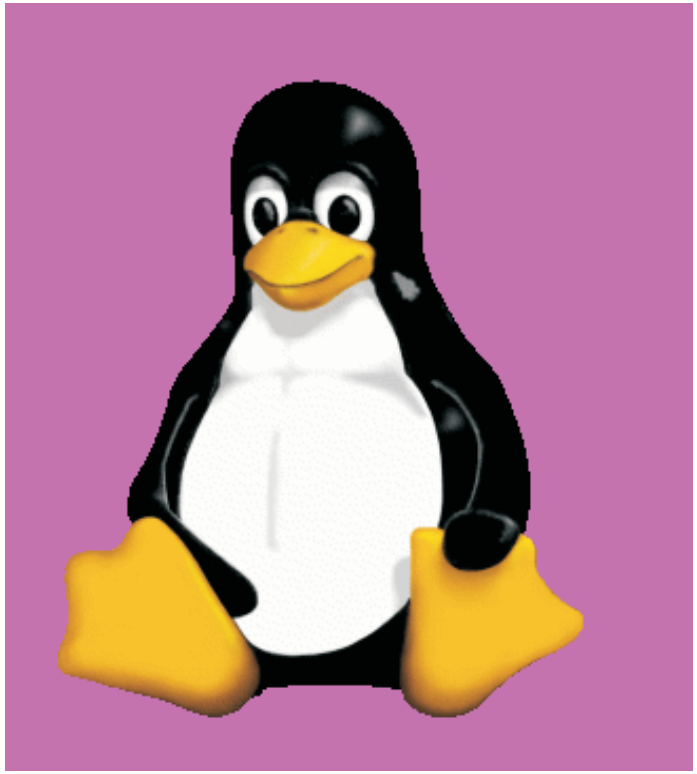
Posted on Internet: October 5, 1991

Linux 1.0 in 1994 and Linux 2.2 in 1999

Today -- runs on 7-10 million computers

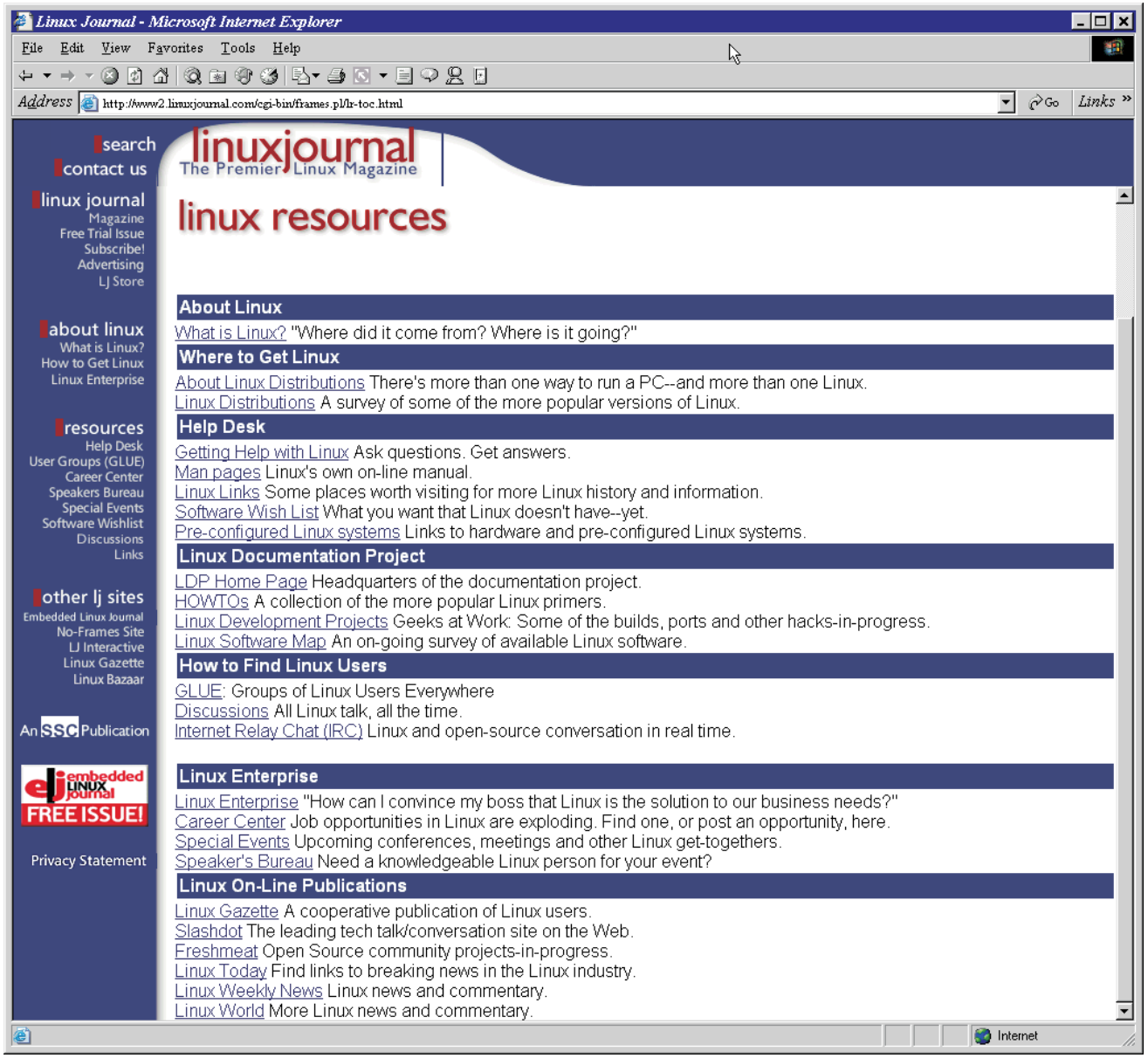
Thousands of programmers

Moving beyond the "enthusiast phase"
into small businesses, ISPs, and
into the corporate IS world



How do you Get Linux

Users of Linux



What's in a Distribution?

Linux kernel

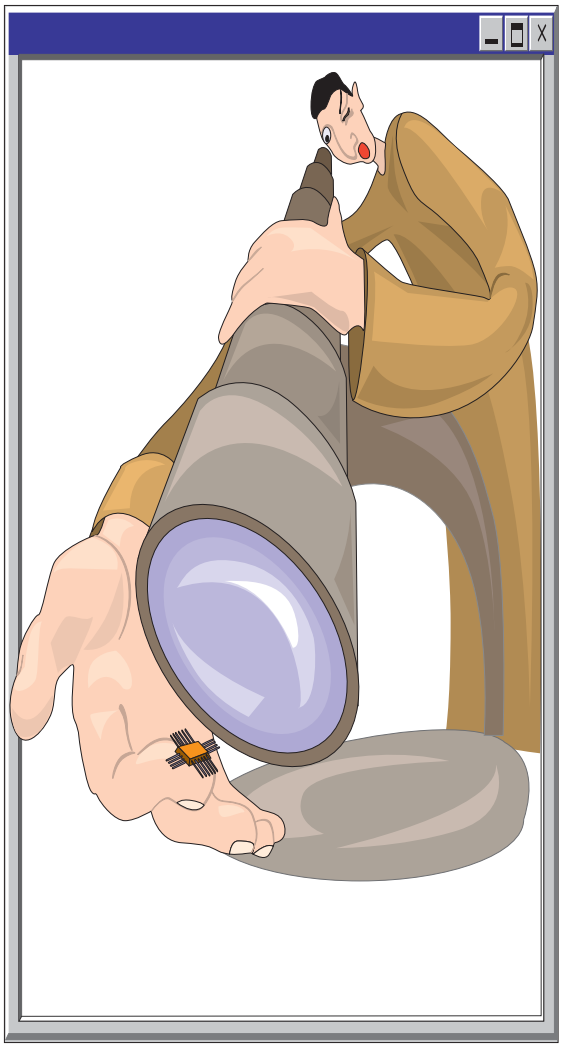
X Window system and window Managers like GNOME and KDE

Web servers, e-mail servers, FTP server

Installation & system configuration support

Third-party applications

Development tools



UNITEDLINUX

SuSE

CALDERA systems

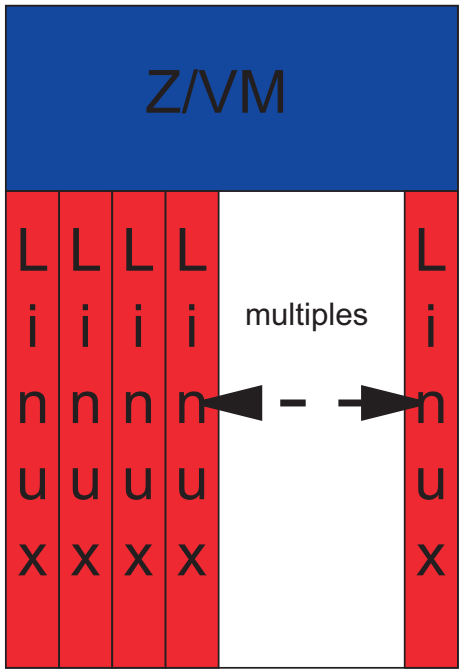
TURBOLINUX

redhat

What's in an Enterprise Distribution?

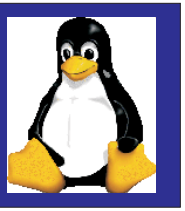
Key Features
2.4.9 kernel (31 bit)
gcc 2.95.3
glibc 2.2.4
ext3 journaling file system
large file support with SW RAID
support for lcs network interfaces

Supported Hardware
IBM S/390 Parallel Enterprise Server
G5: 9672-Rn6, 9672-Yn6
G6: 9672-Xn7, 9672-Zn7
IBM Multiprise 3000 Server - 7060-nnn
IBM zSeries 900 and 800 servers (in 31 and 64 bit mode)



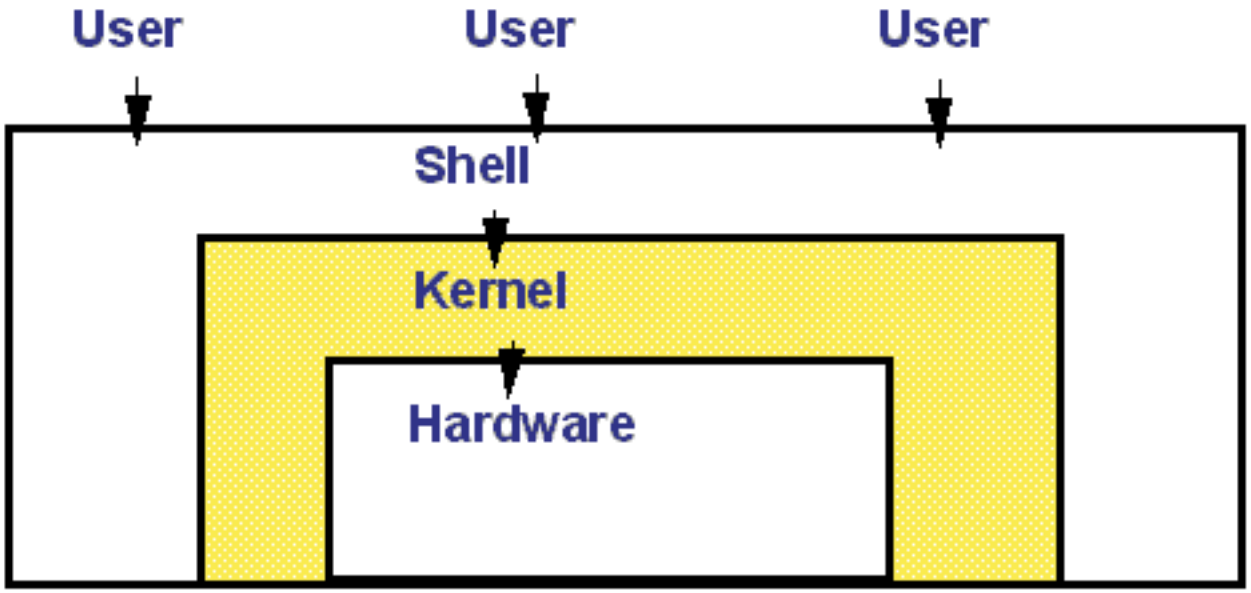
Minimum Requirements
Memory - 64 MB minimum (128 MB recommended)
Disk - 2 DASD partitions of at least 2 GB each

Linux Structure

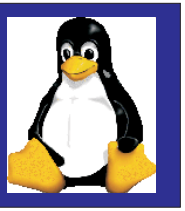


UNIX-like

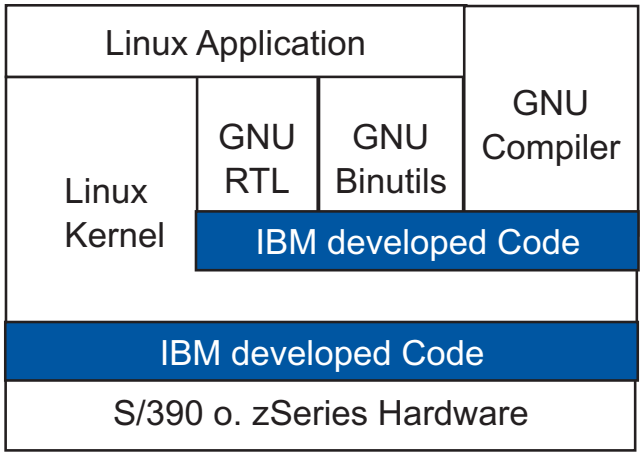
- **kernel:** the part of the operating system that interacts with the hardware
- **shell:** the part of the operating system that interacts with the user



Linux Structure on the Mainframe



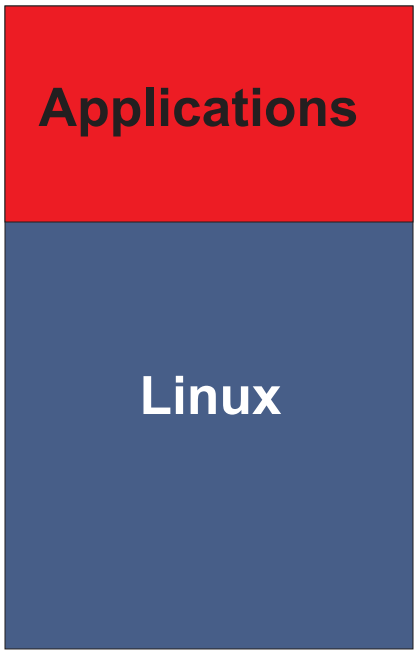
- **Pure Linux, an ASCII environment**
- **Exploits IBM zSeries and S/390 hardware, including IEEE floating point, HiperSockets, ...**
- **Design Principals**
  - No changes on Linux - not a unique version of Linux
  - No changes on Look & Feel for Linux on zSeries
  - No replacement for any other IBM ^ operating system



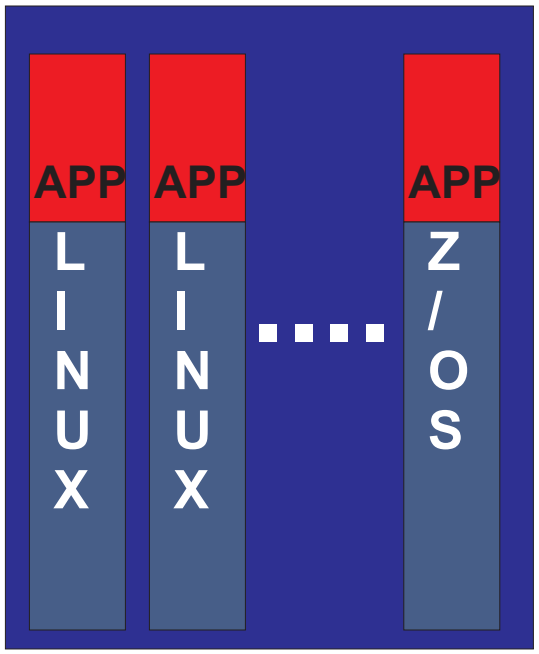
Delta vs. Linux on other platforms < 1%

Linux is Linux

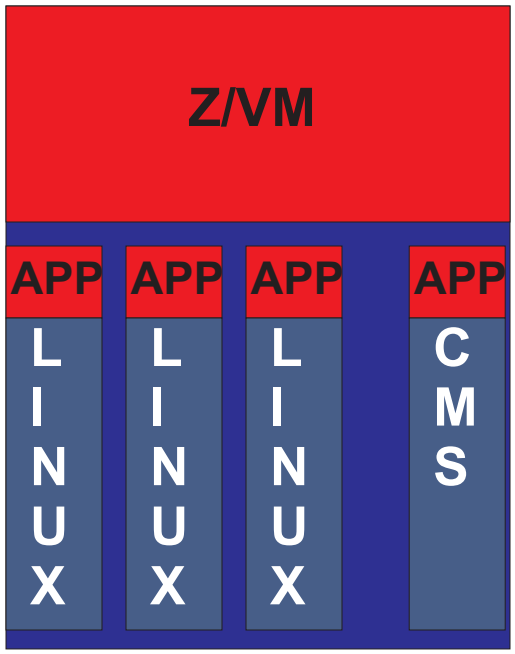
Linux Mainframe Implementations



Zseries Native



Zseries LPAR
(Up to 15 LPARs)



Zseries z/VM
Guest

Agenda

Background

Installing Linux

Major Components of Linux

Resources

Native zSeries Installation

Build 390 Tapes from distribution Cd's
- Transfer files from desktop to z/OS
- Have userid capable of submitting JCL
- Copy code and create 390 tapes
- Have authority to vary devices on/off
- Have access to a tape drive
- Have access to HMC

http://linux390.marist.edu

SG244-987 Linux for OS/390

1) Build IPLable tape from code
2) Load your Linux partition using tape
3) Format DASD and create file system
4) Uncompress files on file system
5) Customize files on file system
6) Create and activate swap system
7) ReIPL from DASD

Z/VM Installation

1) Decide on installation items
   Reader or tape?
   (Reader is easier)
   Which disk drives to user?

2) Prepare VM to run Linux
   by setting up VM environment

3) Prepare network environment
4) Obtain binary fiels

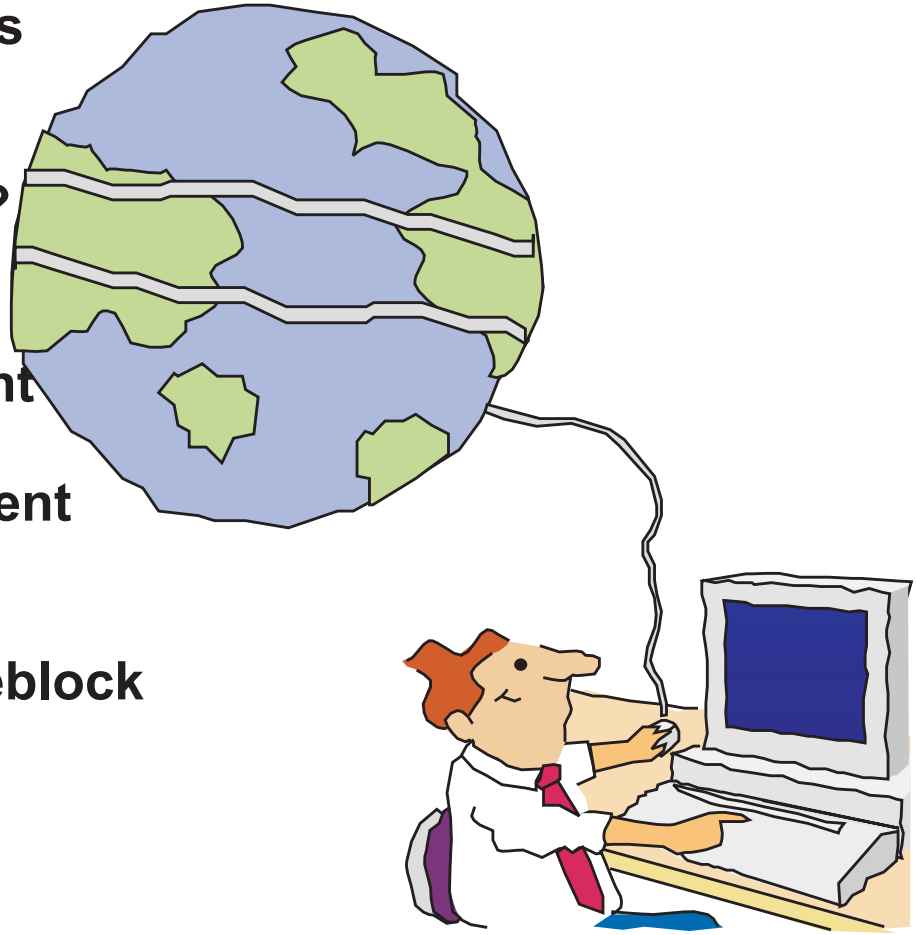
5) Copy the files to VM and reblock

6) Create the initial Kernel

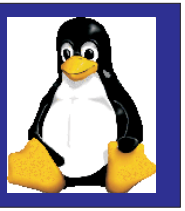
7) Boot the linux kernel

8) Install the root file system

9) Configure



Linux System Administration



Install software

Define user accounts

Configure peripheral devices

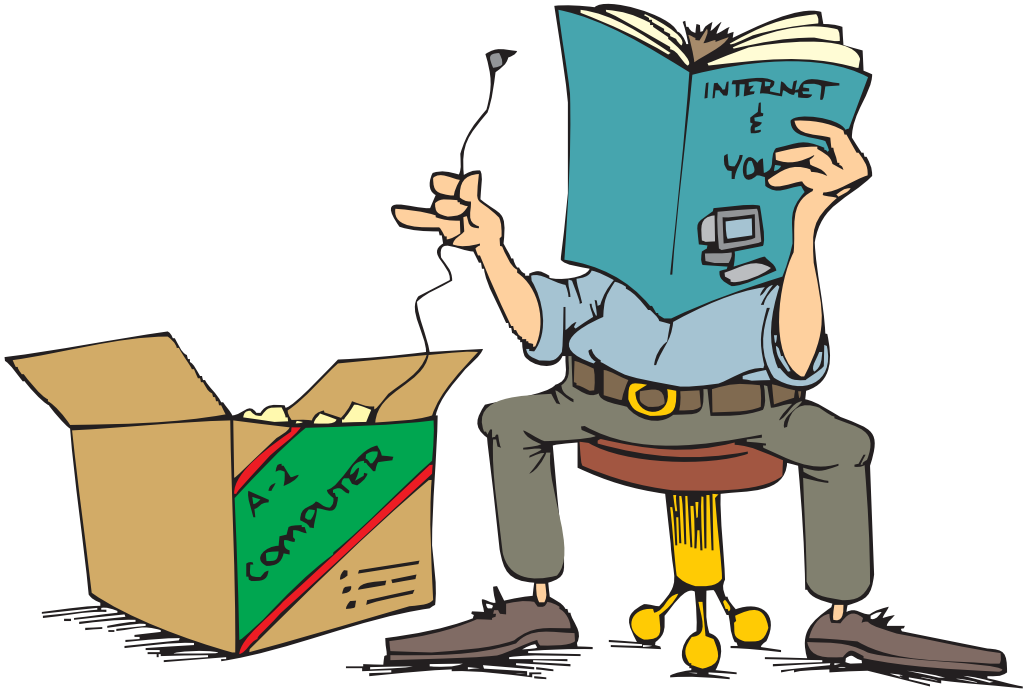
Allocate disk storage

Back up data

Monitor performance

Determine/solve system problems

and so on....



Networking Linux

Configure and support TCP/IP

Use networking commands for
remote log on, remote execution, and file transfer

Do basic troubleshooting of network problems

Support LAN attached printers

Determine Network Problems

Implement NFS

Configure Apache web server

Configure Samba

Agenda

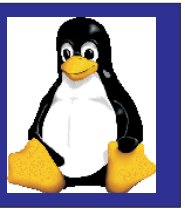
Background

Installing Linux

Major Components of Linux

Resources

Core Components



Kernal

File system

Shell

Network

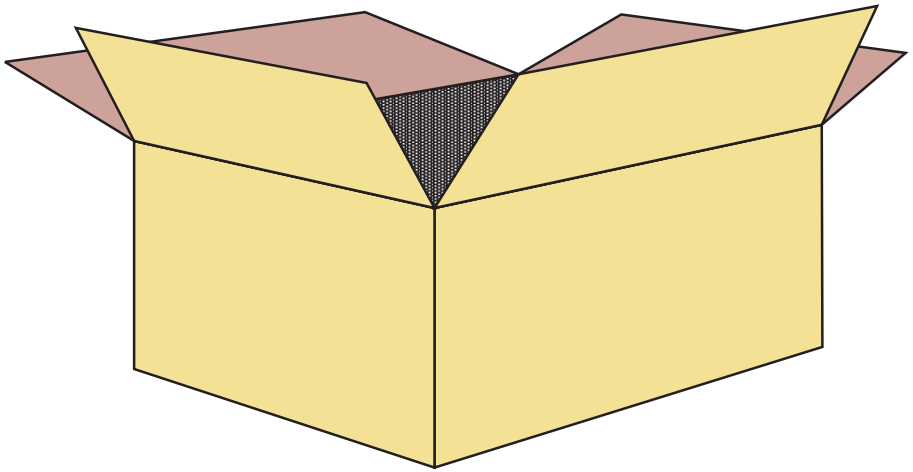
Text processing

Programming

System Mangement

Online documentation

Graphical Interface



We will use the command line interface for most examples

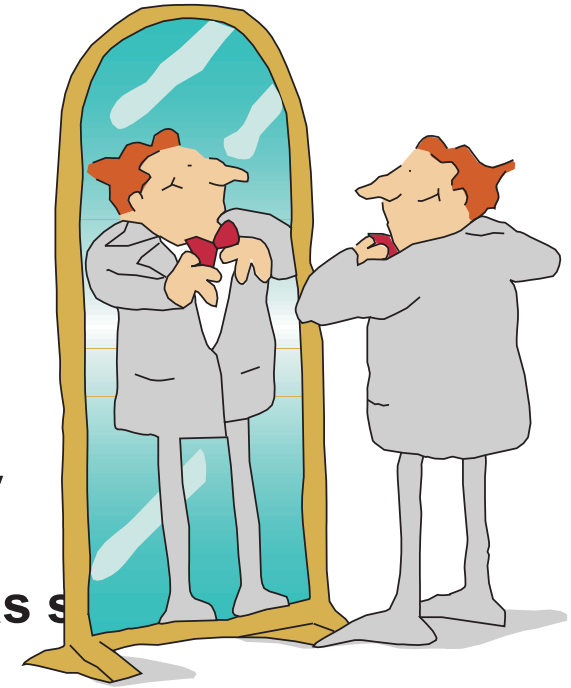
Login and Logout

Telnet to the new system

telnet 10.10.10.1

You are prompted for userid and password

suse390 login: root
password:



root is an administrative userid - use sparingly

Prompt will appear indicating that the login was s

suse390: #

Logout

 # <ctrl d>
 # exit
 # logout

Shutdown

#shutdown -r now
#shutdown -h now

Common Directory

User

/bin
/boot
/dev
/disks
/etc
/home
/lib
/mnt
/opt
/proc
/root
/sbin
/tmp
/usr
/var

Files can be of type

Ordinary
Text or code data
No particular internal format

Directory
A table of contents
A list of the files within
that directory

Special Files
Represent hardware or
logical devices
Found in directory called /dev
For example: /dev/lp0

File Permissions

For an ordinary file:

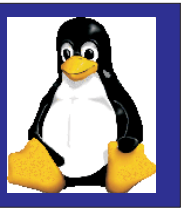
r	read	can look at the contents of the file
w	write	can change or delete the contents of a file
x	execute	can execute the file as a command (r also needed)

For a directory:

r	read	can find out what files are in directory
w	write	can create and remove files from directory (x also needed)
x	execute	has the permission to be in directory

```
drwxrwxr-x  2 team01 staff  1024  Aug 12  10:16  c
-rwxr-xr-x  2 team01 staff  1024  Feb 18  09:55  doc
```

Changing File Permissions



File and directory permissions can be specified in the symbolic syntax or as an octal number:

chmod mode filename

[a u g o] [+ = -] [r w x]
a = all, u = owner, g = group, o = others
+ = add, = = clear permissions,
- = remove permission
r = read, w = write, x = execute

	User	Group	Other
Symbolic notation	rwx	rw-	r-x
Binary	111	110	101
	4+2+1	4+2+0	4+0+1
Octal	7	6	5

Setting file permissions drives everyone crazy

Command Format

Command Option(s) argument(s)

#mail -f mailbox
#passwd
#pwdadm

While graphical interfaces now resemble windows.....many
Linux administrators use the command interface

Keyboard shortcuts
<ctrl-c> terminates command
<ctrl-d> end of line
<ctrl-s> terminates output to the screen
<ctrl-q> resumes output to the screen
<ctrl-u> erases the entire line
<arrow up> previous command
<shift page up> look at output of previous commands

Documentation and Information

There is no accepted directory structure between the distributions
Expect differences (even between releases)

Use the *man* command to read the page of commands

Manual pages are normally stored in */usr/man*

Most files stored in ‘gz’ or ‘tar’ format
- compressed files

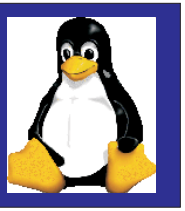
Info command is an alternate way to read manuals

Information stored in */user/info*

Frequently asked questions (FAQ)

Information stored in */user/doc/faq*

vi editor



Full-screen editor

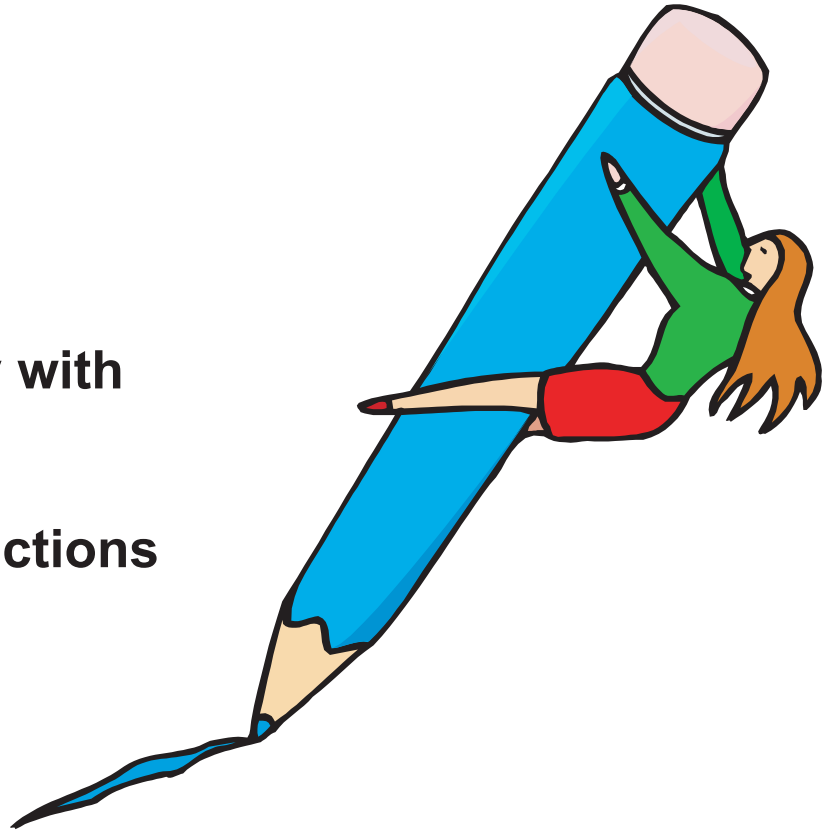
Two modes of operation
- **command**
- **text**

Utilizes one-letter command
Does not format text

Flexible search and replace facility with pattern matching

Allows for user-defined editing functions using macros

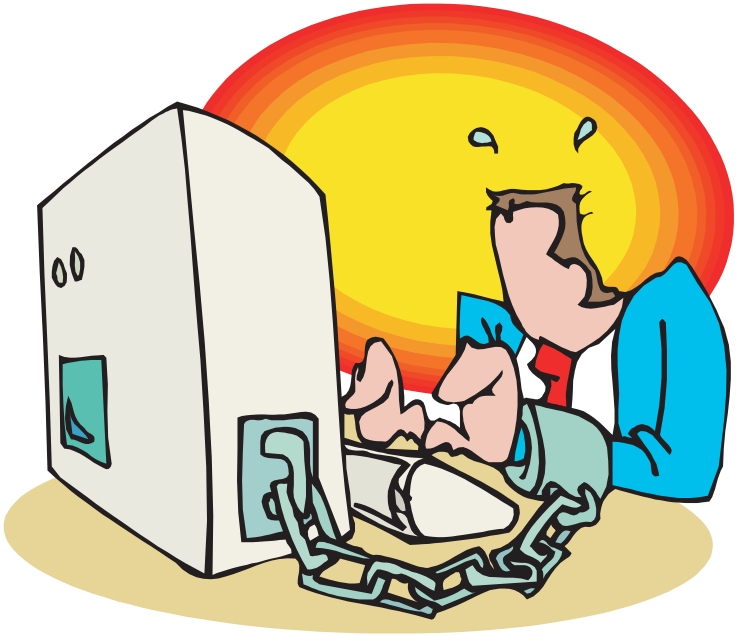
#vi filename



type :help iccf for information
type :q to exit
type :help online help
type :help version6 for version information

vi editor Commands

Options entered in the vi session change the behavior of the vi editor:

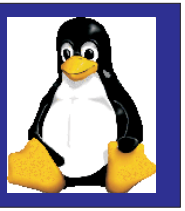


:set all
:set autoindent/noautoindent
:set number/nonumber
:set list/nolist
:set showmode/noshowmode
:set tabstop=x
:set ignorecase/noignorecase
:set wrapmargin=x

Options can be stored in $HOME/.exrc
Macros can be written and new commands created

To delete a single character	**x**
To delete to the end of the current word	**dw**
To delete to the end of the lined	**$**
To delete to the start of the lined	**0**
To delete the whole lined	**d**
To delete a range of lines	**:20,40d**
Replace text by overtyping	**Rnewtext**

The Shell



User interface to Linux
Command interpreter
Enables multiple tasks
Comprehensive programming language

Default in Linux: bash (Bourne Again Shell)

Other shells available: csh, tcsh, pdksh, ash, sash, zsh

Metacharacters are characters that the shell interprets as having a special meaning.

Examples:
< > | ; ! ? * [ ] $ \ " ' ` ~ ( )

Wildcards are a subset of metacharacters that are used to search for and match file patterns.

Examples:
? * [ ] [-]

Many users now using windows-like graphical interfaces

Controlling Processes

The ps command displays process status information

```
# ps jf

PPID PID ... TTY STAT UID TIME  COMMAND
   1 374 ...   1 S    500 0:00  -bash
 374 569 ...   1 S    500 0:00  \_ bash
 569 572 ...   1 R    500 0:01  \_ find /
 369 575 ...   1 R    500 0:00  \_ ps jf
```

ps has a number of command line options:

a shows all processes
u shows user names instead of UID
x kernel processes

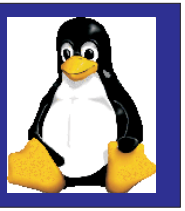
Foreground Processes

#ls -l Invoked by simply typing a command at the command line.

Background Processes

#ls -l & Invoked by putting an "&" at the end of the command line

Controlling Processes



Foreground process

ctrl-c Interrupt key, cancels a foreground process. After the interrupt, the system returns the prompt on the screen

kill Sometimes the kill command is used to terminate foreground processes

fg resume suspended task in the foreground

Background process

kill The kill command is the only way to terminate background processes

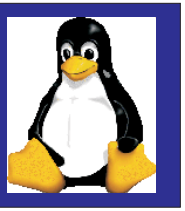
<ctrl-z> suspends foreground task

job slists background or suspended jobs

bg resume suspended task in the background

Specify a job number for bg, fg and kill using %job

Shell Defaults



/etc/profile
/etc/bashrc
#HOME/.bash_profile
#HOME/.bashrc
#HOME/.bash_logout

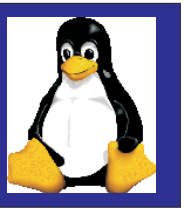
Sample /ert/profile

```
PATH=$PATH:/usr/X11R6/bin
PS1="[\u@\h \W]\\$ "

if [ `id -gn` = `id -un` -a `id -u` -gt 14 ]; then
  umask 002
else
  umask 022
fi

USER=`id -un`
MAIL="/var/spool/mail/$USER"
export PATH PS1 USER MAIL
```

Shell bash Samples



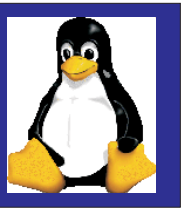
Sample bash_profile

```
# Get the aliases and functions
if [ -f ~/.bashrc ] ;
then
 .~/.bashrc
fi
# User specific environment and startup programs
PATH=$PATH:$HOME/bin
BASH_ENV=$HOME/.bashrc
export PATH BASH_ENV
```

Sample bashrc

```
# User specific aliases and functions
alias lsd='ls -FlA | grep ^d'
alias lsf='ls -lA | grep -v ^d'
alias lst='ls -lAt | head'
alias history='\history 10'
alias r='fc -s'
set -o vi
# Source global definitions
if [ -f /etc/bashrc ] ;
then
 ./etc/bashrc
fi
```

Linux Utilities



The find command is used to recursively search directories for files with particular characteristics

The grep command is used to select entire lines containing a particular pattern

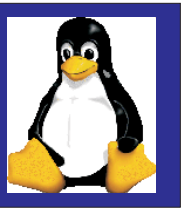
The head and tail commands are used to view specific lines in a file

The sort command sorts the contents of a file by the options specified

Find out where you can find commands with type, which and whereis

The gzip, zcat and gunzip commands can be used to create and work with compressed files

Linux Compression



#ls -l file1
-rw-rw-r-- 1 team01 team01 32031 Apr 6 23:40 file1

 # gzip -v file1
file1:89.9% -- replaced with file1.gz

 # ls -l file1.gz
-rw-rw-r-- 1 team01 team01 3265 Apr 6 23:40 file1.gz

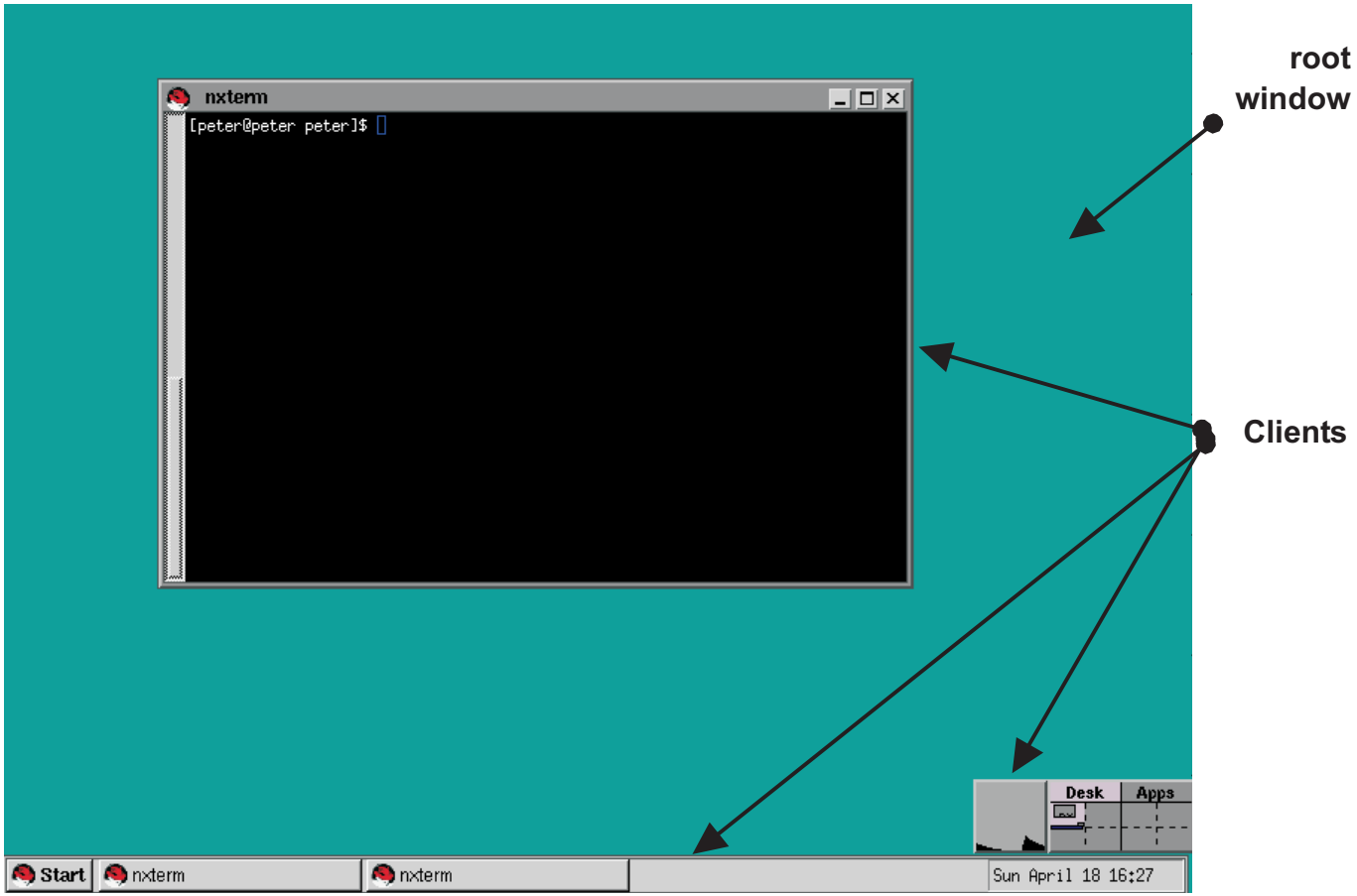
 # zcat file1
(output is the same as the output of the cat command with the uncompressed file)

 # gunzip file1
 # ls -l file1
-rw-rw-r-- 1 team01 team01 32031 Apr 6 23:40 file1

Linux Xwindows

Network-based graphics system developed at MIT in 1984
Freely available form the X-Consortium
XFree86 in the X Window implementation used in Linux
Enables you to run graphical applications

$ startx



Linux Xwindows

Each X Server:
Controls one keyboard, one mouse and one or more screens
Allows simultaneous access by several clients
Performs basic graphic operations
Provides information such as fonts and colors
Routes keyboard and mouse input to the correct clients

X clients are the applications themselves which the user runs under the X Window system

Some examples of X clients are programs such as:
nxterm, xterm, xclock, xcalc, xman, xedit, xlsfont

X clients can be started from the command line of from special startup files

Most X clients share the same options for specifying attributes such as foreground color, background color, display name, window geometry and font

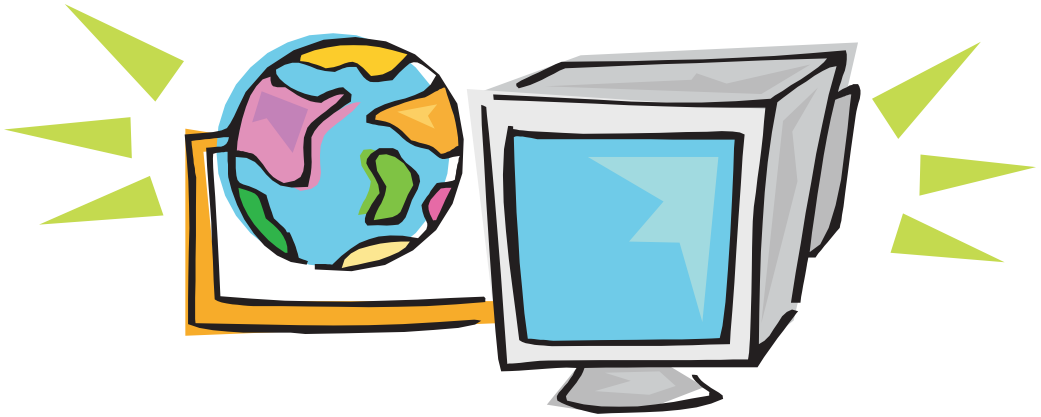
LinuxConf

Powerful system administration tool - must be root to use

Created by Jacques Gelinas
http://www.solucorp.qc.ca/linuxconf

Uses tree structure to change configuration data

Four interface modes:
Command line
  Useful when writing scripts
Character-cell
  Interactive, text based
X Window based
  Interactive, GUI
Web-based
  For remote system administration



Logged to /var/log/netconf.log

LinuxConf Command Line

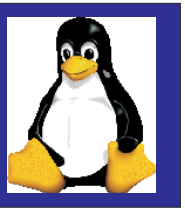
Very practical in scripts

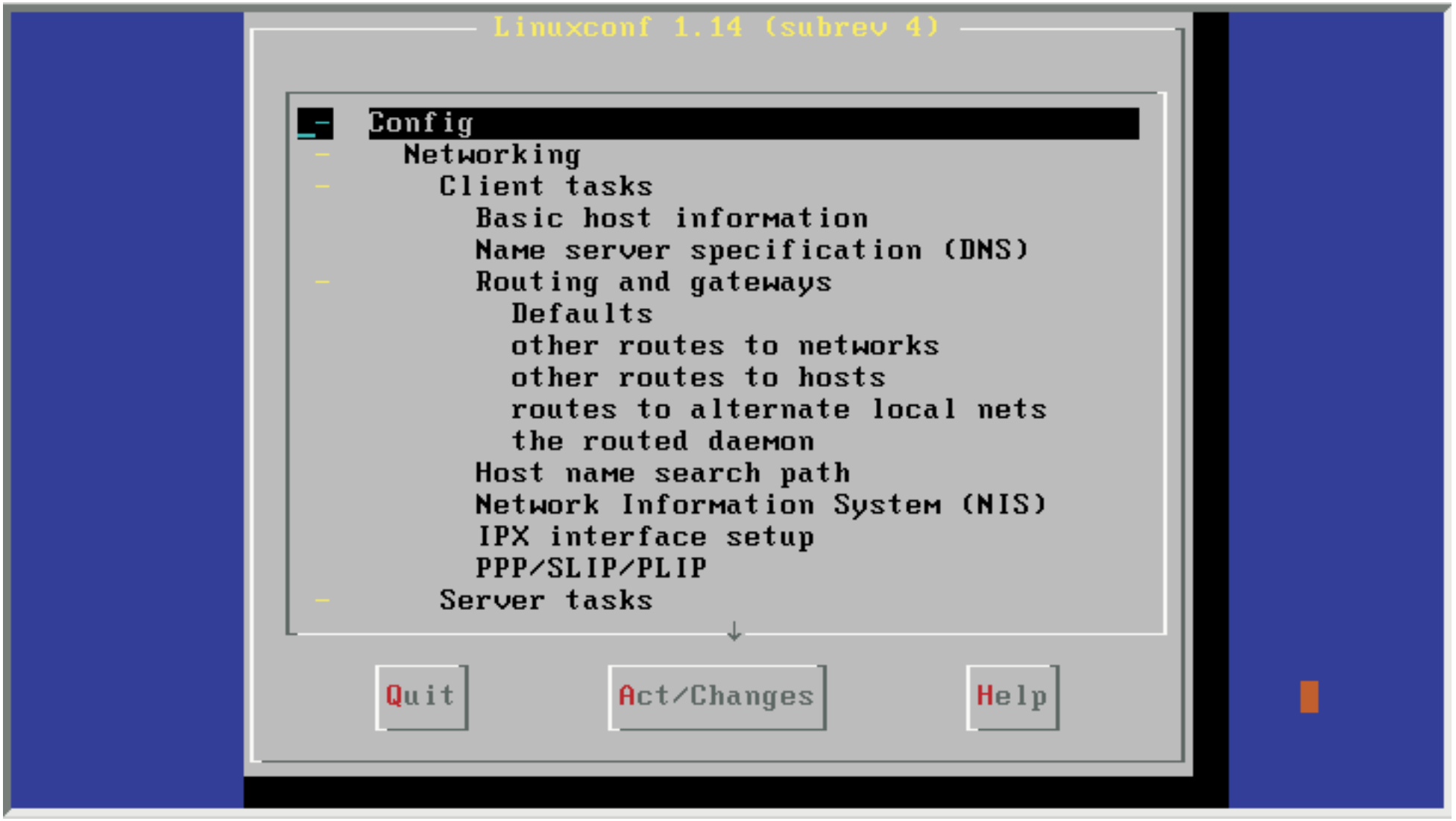
Examples:

```
# linuxconf --status
# userconf --adduser tux1 tux1
 "Tux" /bin/bash
# netconf --setgateway 10.0.0.1
# fsconf --check
```

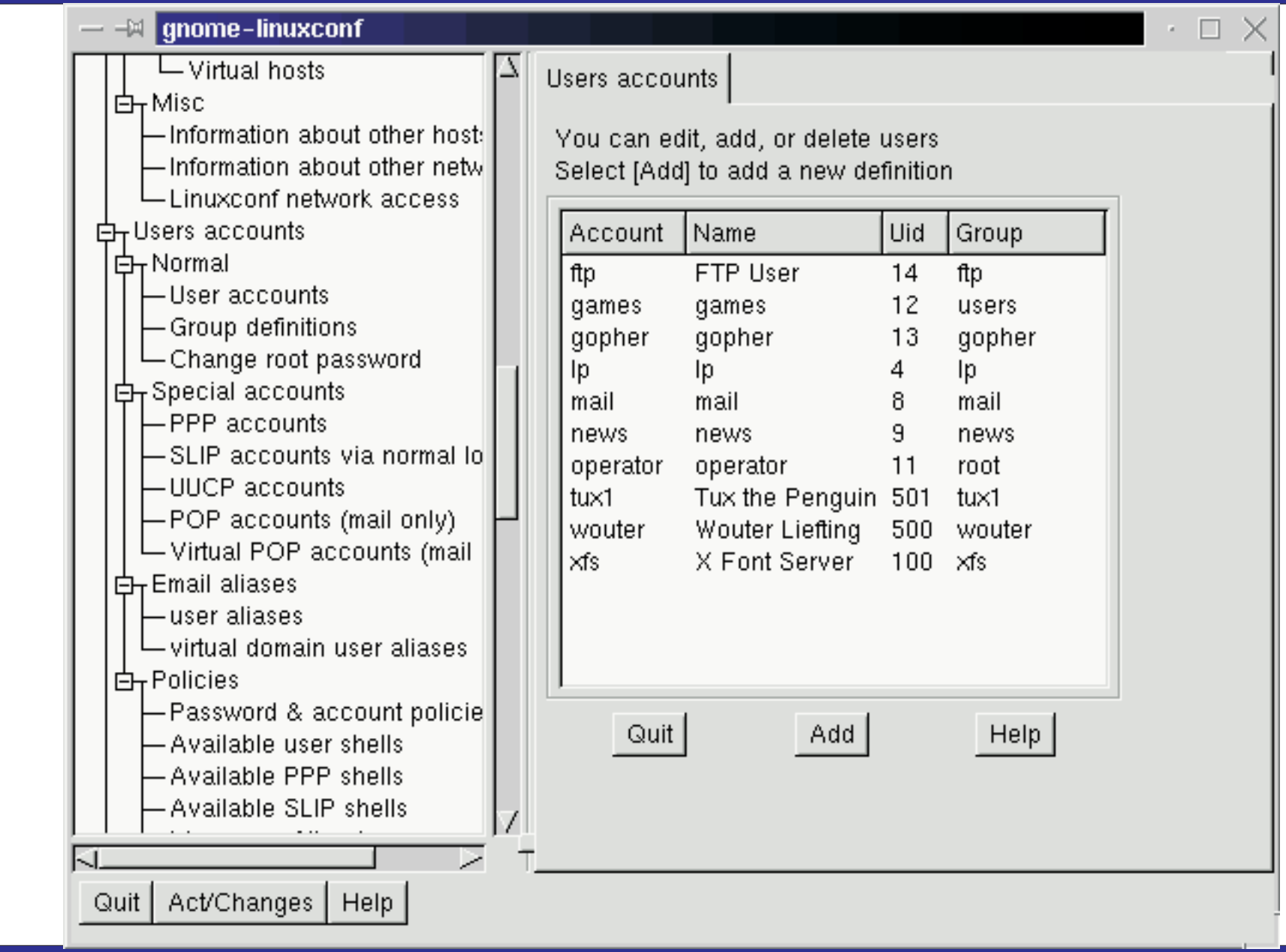
```
# cd /bin
# ls -l fsconf userconf netconf
lrwxrwxrwx 1 root root 14 Jul 15 16:36 fsconf -> /bin/linuxconf
lrwxrwxrwx 1 root root 14 Jul 15 16:36 netconf -> /bin/linuxconf
lrwxrwxrwx 1 root root 14 Jul 15 16:36 userconf -> /bin/linuxconf
```

LinuxConf Cell Interface





LinuxConf Xwindows Interface



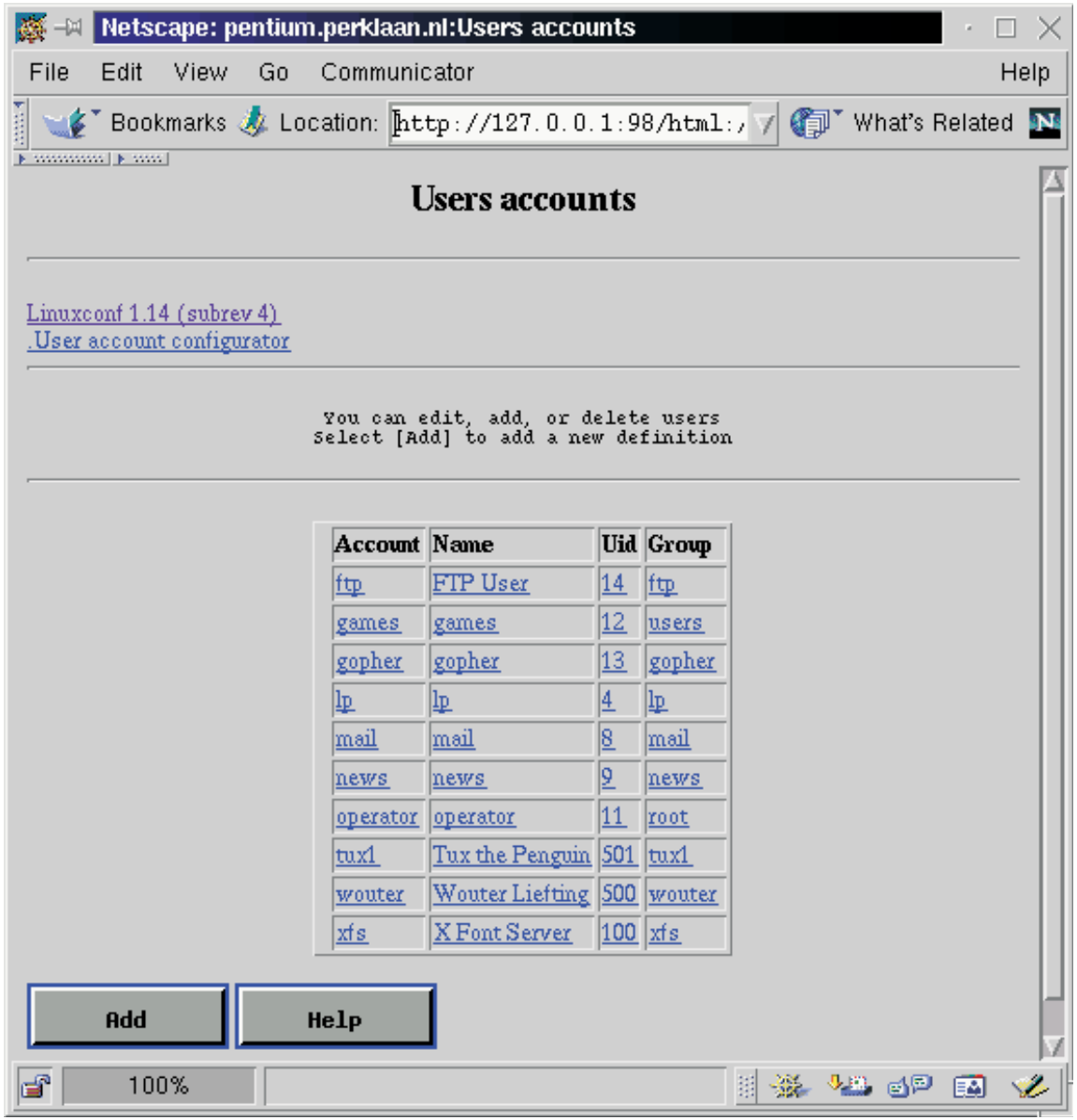
LinuxConf Web Interface

Disabled by default

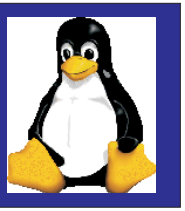
To enable:

Config; Networking; Misc; Linuxconf network access

To access:
Enter http://<host>:98/ as URL in any browser

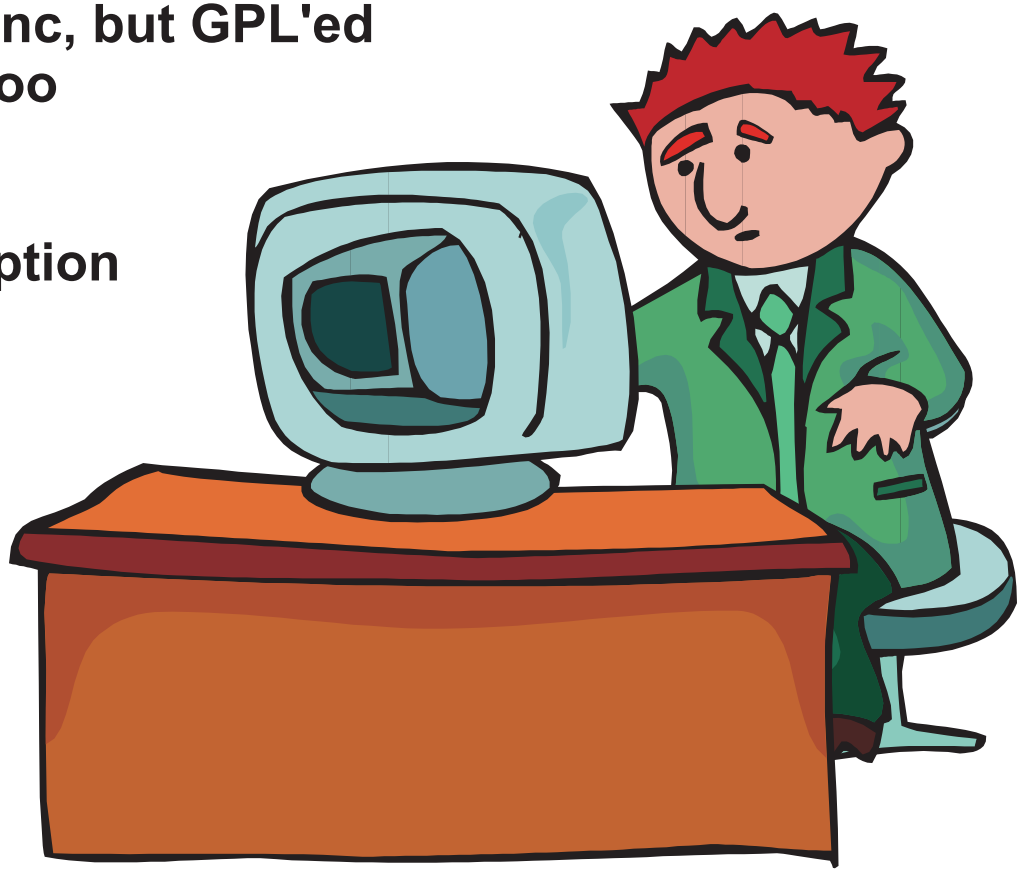


Red Hat Package Manager (RPM)



Used to install/deinstall packages with .rpm file

Developed by Red Hat Software Inc, but GPL'ed
Other Linux distributions use it too

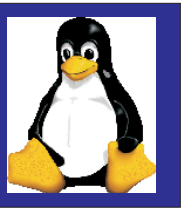


Uses .rpm files which contain
- Package name, version, description
- Dependency information
- The program files itself
- Pre- and post install scripts

RPM database (/var/lib/rpm) contains database of installed packages

Five basic modes
- Installing
- Freshening and Upgrading
- Uninstalling
- Querying
- Verifying

RPM Command Line



Installs, freshens or upgrades an RPM

Freshen: only install if an older RPM was installed

Upgrade: always install, but uninstall older RPM first

Basic syntax:

```
rpm -i package-filename.rpm
rpm -F package-filename.rpm
rpm -U package-filename.rpm
```

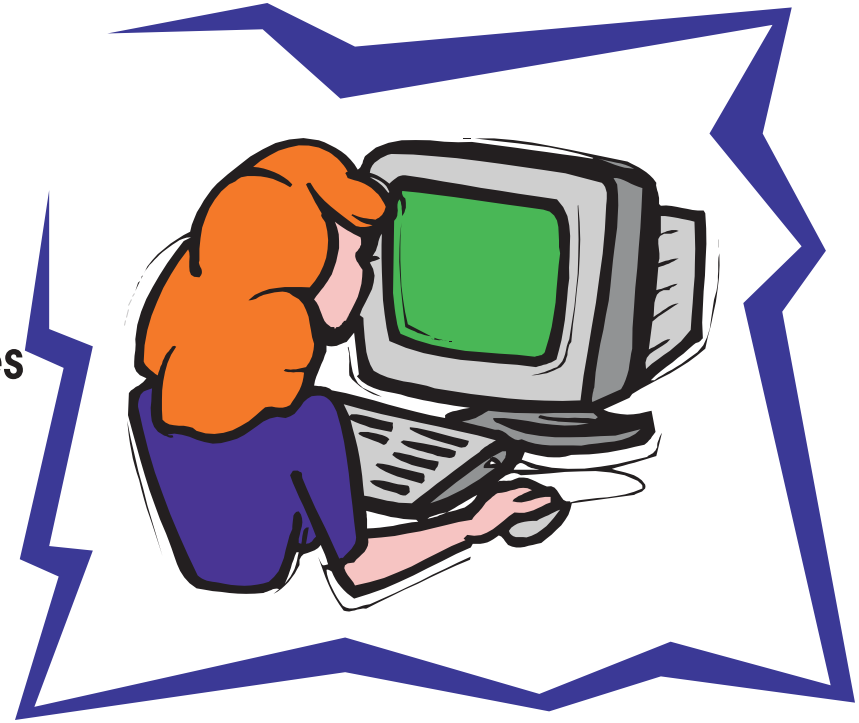
-vverbose

-h print 50 hash marks

-nodeps don't check dependencies

When upgrading, old configuration files are saved with extention .rpmsave

Package-filenames may also be specified as URLs



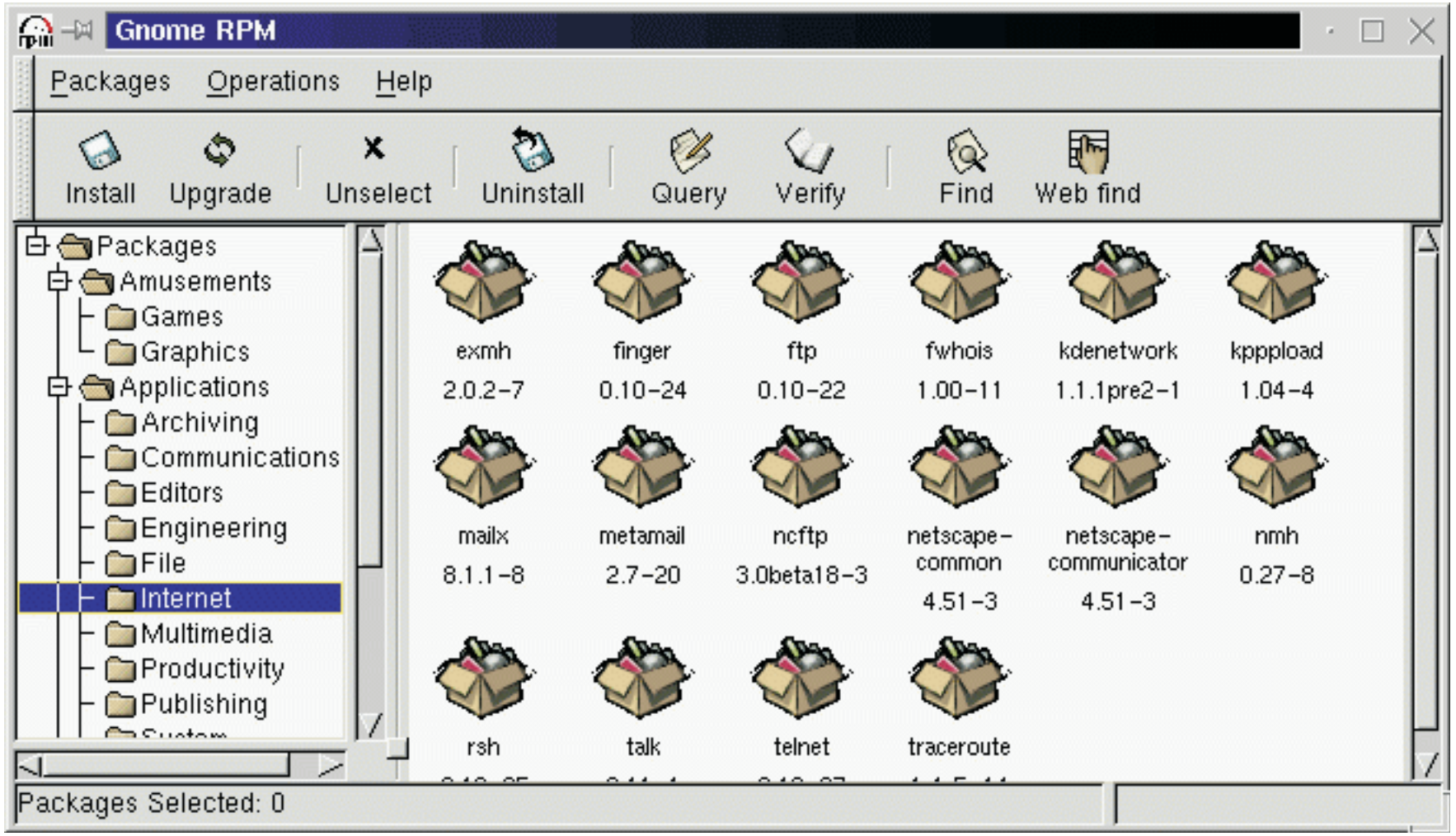
Uninstalls an RPM

Basic syntax:

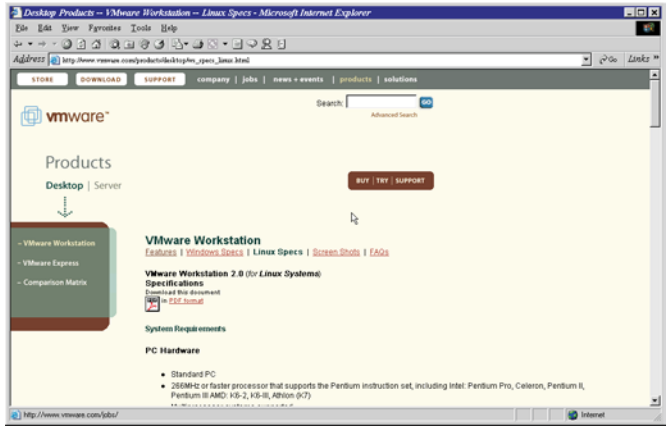
```
rpm -e package-name
```

–nodepsignore any dependency breaks

RPM Graphical Interface



Windows Emulation



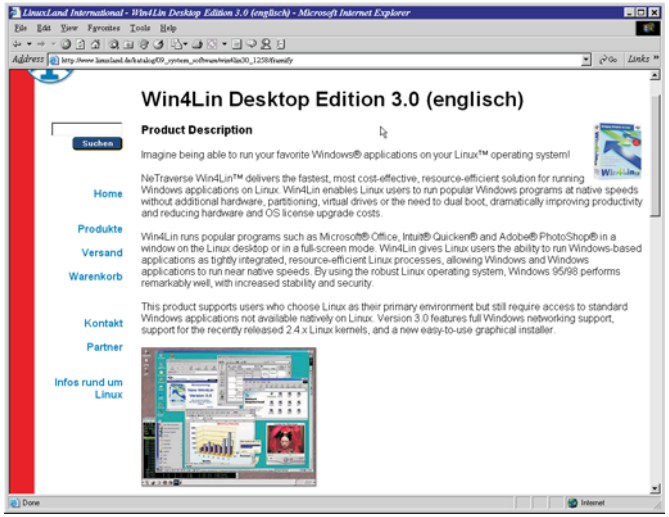
dosEMU runs older DOS applications.

WABI environment runs 16-bit MS windows applications

WINE emulator for 32-bit MS windows applications

VMWare delivers a flexible and safe computing environment by providing multiple virtual computers on a single PC

Win4LIN runs 32 bit MS windows applications



Summary

Discussed the history of Linux

Described the options for installation on the mainframe

Investigated some of the basic functions

Don’t expect all distributions to be alike

Lots of options

Interfaces still in flux, but one exists to suite every taste

Mainframe versions designed to support needs of a mainframe world

Agenda

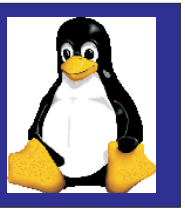
Background

Installing Linux

Major Components of Linux

Resources

Resources



Kernel source : ftp.kernel.org, ftp.funet.fi
Network tools: ftp.uk.linux.org
Network applications: ftp.uk.linux.org
Linux Networking: secretagent.com/doc/howto/NET-3-HOWTO-6.html
Mailing list:
To: majordomo@vger.rutgers.edu
Subject: Subscription
Message: subscribe linux-net
Linux Networking newsgroup:
comp.os.linux.networking
Linux online: www.linux.org
Linux documentation:
metalab.unc.edu/mdw/index.html
Linux international : www.li.org
Linux businesses: www.linux-business.com
Linux resources: www.linuxresources.com
General: www.linux.com
Linux journal: www.linuxjournal.com
Linux news: linuxtoday.com
Open Source Community: slashdot.org
Linux software: www.linuxbase.com