Unix/IP Preparation Course
May 6, 2012
Serrekunda, The Gambia
# Ubuntu Timeline

<table>
<thead>
<tr>
<th>Version</th>
<th>Code name</th>
<th>Release date</th>
<th>Supported until Desktops</th>
<th>Supported until Servers</th>
<th>Colour</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.10</td>
<td>Warty Warthog</td>
<td>20 October 2004</td>
<td>30 April 2006</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.04</td>
<td>Hoary Hedgehog</td>
<td>8 April 2005</td>
<td>31 October 2006</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.10</td>
<td>Breezy Badger</td>
<td>13 October 2005</td>
<td>13 April 2007</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.06</td>
<td>Dapper Drake</td>
<td>1 June 2006</td>
<td>14 July 2009 1 June 2011</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.10</td>
<td>Edgy Eft</td>
<td>26 October 2006</td>
<td>25 April 2008</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.04</td>
<td>Feisty Fawn</td>
<td>19 April 2007</td>
<td>19 October 2008</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.10</td>
<td>Gutsy Gibbon</td>
<td>18 October 2007</td>
<td>18 April 2009</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.04</td>
<td>Hardy Heron</td>
<td>24 April 2008</td>
<td>12 May 2011 April 2013</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.10</td>
<td>Intrepid Ibex</td>
<td>30 October 2008</td>
<td>30 April 2010</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.04</td>
<td>Jaunty Jackalope</td>
<td>23 April 2009</td>
<td>23 October 2010</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.10</td>
<td>Karmic Koala</td>
<td>29 October 2009</td>
<td>30 April 2011</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10.04</td>
<td>Lucid Lynx</td>
<td>29 April 2010</td>
<td>April 2013 April 2015</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10.10</td>
<td>Maverick Meerkat</td>
<td>10 October 2010</td>
<td>April 2012</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11.04</td>
<td>Natty Narwhal</td>
<td>28 April 2011</td>
<td>October 2012</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11.10</td>
<td>Oneiric Ocelot</td>
<td>13 October 2011</td>
<td>April 2013</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12.04</td>
<td>Precise Pangolin</td>
<td>26 April 2012</td>
<td>April 2017(130)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Image courtesy of Wikipedia
Unix and Linux

Are they the same?
Yes, at least in terms of operating system interfaces
Linux was developed independently from Unix
Unix is much older (1969 vs. 1991)

Scalability and reliability
Both scale very well and work well under heavy load
(this is an understatement😊)

Flexibility
Both emphasize small, interchangeable components

Manageability
Remote logins rather than GUI
Scripting is integral

Security
Due to modular design has a reasonable security model
Linux and its applications are not without blame
UNIX / Linux and Windows

Why does AfNOG use UNIX / Linux?

It’s what the Internet uses to provide core services
60% of all web servers are running Apache
Much of Enterprise class computing built around UNIX / Linux
Open Source network monitoring and management solutions
- Widely used
- Generally not available for Windows

Router OSes are command-line and some, even, Linux

We assume
End users are on Windows (some places Macs, too)
Don’t expect end-users to use UNIX or Linux
We do expect that you are likely to use Linux or UNIX

Licensing
Windows products cost $$
Open Source software is “free” (as in beer)
Actual costs to implement vary widely
The Unix System Simplified

- Hardware
- The UNIX Kernel
- The Shell Program
- Command Processes
The Unix System More Detail

Background processing
Non-interactive (no terminal)

Launch processes
User interaction (terminal)
Job control (scripting)

DAEMONS

I/O (+ terminals)
Memory management (VM)
Interrupts
Scheduling / Timesharing
Networking

File management
Editors
Compilers
Network tools
...

KERNEL

Eat Resources :-)
(uid / gid)

SHELLS

KERNEL

APPs

KERNEL

APPs

KERNEL

KERNEL

KERNEL

KERNEL
The Kernel

• The "core" of the operating system
• Contains device drivers
  - Communicate with your hardware
  - Block devices (physical media – hard drive, CD, RAM)
  - Character devices (keyboards, mice, terminals, modems)
  - Network devices (network cards)
  - Pseudo devices (/dev/null, /dev/random)

• Filesystems
  - Organise block devices into files and directories
  - ufs2, ext2, ext3, ext4, reiserfs, jfs, zfs
The Kernel continued

Memory management
  - Real, Virtual and paging algorithms

Timeslicing (multitasking)
  - Resource allocation to processes

Networking stacks - esp. TCP/IP
  - Packets traverse the kernel

Enforces security model
  - Does this user have privileges
  - Numeric userid identifiers ("uid")
  - id 0 is “special” - root
Shells

Command line interface for executing programs
  • Windows equivalent: command.com or command.exe

Also programming languages for scripting
  • DOS/Windows equivalent: batch files, WSF, VBScript

Choice of similar but slightly different shells
  • **sh**: the "Bourne Shell". Standardised in POSIX
  • **csh**: the "C Shell". Not standard, but includes command history
  • **bash**: the "Bourne-Again Shell". Combines POSIX standard with command history.
  • Others: **ksh**, **tcsh**, **zsh**
The programs that you choose to run

Frequently-used programs tend to have short cryptic names (why?)
  "ls" = list files
  "cp" = copy file
  "rm" = remove (delete) file

Lots of stuff included in most base systems
  Editors, compilers, system admin tools

Lots more stuff available to install as well
  Thousands and thousands of packages
Programs that run in the background; also known as "daemons"

Examples:

- **cron**: Executes programs at certain times of day
- **syslogd**: Takes log messages and writes them to files
- **inetd**: Accepts incoming TCP/IP connections and starts programs for each one
- **sshd**: Accepts incoming logins
- **sendmail** (other MTA daemons like Exim, Postifx): accepts incoming mail (smtp)
Security model

Numeric IDs
- user id (uid 0 = "root", the superuser)
- group id
- supplementary groups

Mapped to names in plain text files
- /etc/passwd
- /etc/group

Suitable security rules enforced
- e.g. you cannot kill a process running as a different user, unless you are "root"
Any questions?
Connect to your Virtual Linux Machine

Now you will use ssh to log in on your virtual Linux machine:

2. Save putty.exe to your desktop and double-click the icon
3. Connect to pcN.ws.nsrc.org as user “sysadm”
   We’ll do this now and instructors will help

• Mac / Linux users open a terminal window and do

   $ ssh sysadm@pcN.ws.nsrc.org

   Password for sysadm user will be given in class
Core directory refresher

/ (boot, /bin, /sbin, /etc, maybe /tmp)
/var (Log files, spool, maybe user mail)
/usr (Installed software packages)
/tmp (May reside under “/”)
/home (user’s home directories reside here)

Don't confuse the the “root account” (/root) with the “root” (“/”) partition.
During FreeBSD installation you can choose this option. It creates the following:

- **“/” Small Root partition**
  - this will contain everything not in another partition
    /bin, /sbin, /usr etc.

- **A *swap partition* for virtual memory**

- **/var**
  - For “variable” files, such as logs, mail spools, etc.

- **/tmp**
  - Where temporary files are located

- **/usr**
  - /usr/home contains user directories. This is the largest partition created. In Linux this is just /home.
Many/most Linux distributions will default to:

- “/” Almost every here. Very large partition.  
  - this will contain everything not in another partition  
    /bin, /sbin, /usr etc.
- A swap partition for virtual memory
- /boot  
  - Contains the linux kernel image(s) and associated configuration and bootstrapping files.
Partitioning Issues

/var may not be big enough
/usr contains OS utilities, third-party software
/usr/home contains your own important data
If you reinstall from scratch and erase /home, you will lose your own data
• Everything in “/” is now more common due to RAID. Why? Valid?
• /tmp?
• Others?
• How much swap should you define?
Sample file layout Linux

- `/root`
  - `/boot`
    - The super-user's home directory
    - The kernel image is in here
    - System configuration files
    - Users' directories are under here
  - `/home`
  - `/mnt`
  - `/sys`
  - `/proc`
  - `/dev`
  - `/bin`
  - `/sbin`
  - `/lib`
  - `/usr`

- `/etc`
- `/sys`
- `/bin`
- `/lib`
- `/usr`

- General purpose mount point
- A view of internal kernel data
- The kernel's view of the hardware
- Special device files live here
- Libraries
- Executables
  - `bin`
  - `sbin`
  - `lib`
  - More executables
  - More libraries
Partitioning is just a logical division

If your hard drive dies, most likely everything will be lost.

If you want data security, then you need to set up mirroring with a separate drive. Another reason to keep your data on a separate partition, e.g. /u

Remember, “rm -rf” on a mirror works very well.

Or, as always “Data Security” <=> Backup
Any questions?
Software installation in FreeBSD:

- Install from source
- Install from binary
- Compile from source using a port
- Use a wrapper tool, such as `portinstall`.
- Install pre-built FreeBSD packages using `pkg_*`

You can keep the source tree local and up-to-date. This is known as the *ports collections*. A number of tools to do this, including `portsnap`. 
Two major packaging systems:
- Redhat Package Manager ➔ RPM
- Debian Packages ➔ DPKG

Both have wrapper tools to make them easier to use:
- rpm wrapped with “yum”
- dpkg wrapped with “apt” and “aptitude”

Both use repositories.

Linux has the other usual suspects as well:
- Install from source
- Install from binary
System Startup FreeBSD

Startup scripts in FreeBSD

- `/etc/rc.d` – system startup scripts
- `/usr/local/etc/rc.d` – third-party startup scripts

Controlling services

- `ln /etc/defaults/rc.conf` – initial defaults
- `/etc/rc.conf` – override settings here
System Startup Linux

Startscripts

In /etc/init.d/ (System V)
In /etc/init/ (Ubuntu 12.04 LTS and Upstart)

NOTE! Upon install services run!

Controlling services

- `update-rc.d` (default method)
- Stop/Start/Restart/Reload/Status Services

  
  # service <Service> <Action>

  or, “old school”

  
  # /etc/init.d/<service> <action>
## Administration

- The use of the *root* account is discouraged. The *sudo* program is used instead.
- You can do a "*buildworld*” to move between major and minor releases (FreeBSD).
- You can use *apt* and/or *yum* to move between many major and minor Linux releases.
- Ubuntu does *[do-release-upgrade]* to move to a new version.
Important Reads

- man builtin
- man hier
- man man
- man ports
- man rc.conf

And, “man any_unknown_command” when you are in doubt.
# There's More

<table>
<thead>
<tr>
<th>FreeBSD Handbook</th>
<th>Ubuntu Resources</th>
</tr>
</thead>
</table>
Packages & Exercises

We'll reinforce some of these concepts using exercises...