Teaching IP networking fundamentals in resource constrained educational environments

Grenville Armitage  
Warren Harrop  
(Jason But)  
{wazz,garmitage,jbut}@swin.edu.au


Outline

• Introduction
• Teaching unix - options
• What is “RULE” (Remote Unix Lab Environment)?
  • For students
  • For teachers
• RULE v2
• Conclusion
Introduction – the problem

• Pedagogical goals
  • Expose students to unix
  • Allow self directed access (flexible learning)
    • Allow access from anywhere on the campus network

• Limitations
  • Existing Windows labs
  • Floor space

Why use unix?

• Important in teaching Telecomms Engineering
  • Ability to modify network code implementation
  • Unix is typically very strong in network centric installations
  • “Hands-on experience” due to many unix-like operating systems being open-source

• Server operating system of choice
  • Unix is the most common OS for Apache web server, the most popular web server in the world
  • Unix is incredibly reliable (netcraft.com)

• Give a range of experiences
What is RULE?

(Remote Unix Lab Environment)

“Giving student(s) a (group of) PC(s) that can be accessed anywhere/anytime for advanced telecoms and computer science teaching”
Deploying unix access – options

Lots of students needing flexible access

Saves space

Wastes space

Central System

Distributed System

Saves space

Wastes space

Central System

Distributed System

Dedicated unix labs

• Multiple root acc’nt
• Kernel access
• Good speed
• Costs
• Tech support (reimaging)
Deploying unix access – options

Lots of students needing flexible access

Saves space

Central System

Single Server

• Single root acc’nt
• Limited student exp admin tasks
• No kernel access

Wastes space

Distributed System

Dedicated unix labs

• Multiple root acc’nt
• Kernel access
• Good speed
• Costs
• Tech support (reimaging)

Virtual Host (RULE)

• Good speed
• Multiple root acc’nt
• No kernel access

Virtual Machines

• Slow or $$
• Multiple root acc’nt
• Kernel access

Using unix – background

Regular computer

User

Kernel

Hardware

Other solutions

Virtual Machines
(Emulation of CPU and hardware devices)

FreeBSD “jails”

Emulated

Hardware

RULE

Virtual Hosts
(FreeBSD “jails”)

User

Kernel

Hardware

$0

$$ or Slow

Emulated

Hardware

$0

FreeBSD “jails”

Emulated

Hardware

$0

FreeBSD “jails”
Using unix – RULE

For students...

- Unix system access
- Flexible Learning
  - 24hr access
  - Internet-wide access
- Other advantages
  - “Sandbox” environment
    - Allows for experimentation
  - “Drop in” guidance by teaching staff
Teaching experiences

- Deployed within Telecoms Eng at Swinburne
- Lab sessions for HET436 – ’03, ’03, ’05
  - Introduction to unix
  - Configure a http (web) server
    - “thttpd” (tiny httpd)
  - Configure a http (web) proxy server
    - “tinyproxy”
  - Configure a Domain name server (DNS) server
    - Berkeley Internet Name Domain (BIND) - “used on the vast majority of name serving machines on the Internet”

Teaching experiences

- Advantages to having virtual hosts
  - “Drop in” on student’s servers – provide help
    - Overwrite files if needed
  - Automated “pickup” of work
  - Plagiarism detection
  - GUI management tool (JHT)...
JHT (Jail Host Toolkit) - Admin

- GUI app for administering virtual hosts

RULE v2
Why RULE v2 ?

• Returning to our original goals...
  “Giving student(s) a (group of) PC(s) that can be accessed anywhere/anytime for advanced telecoms and computer science teaching”

• Virtual host does not provide full functionality
  • As might be required for certain scenarios including telecomms and OS design – cannot modify OS

• If this was made possible
  • Students can now break the OS
  • Potential to create runaway system that generates large amounts of network traffic

Deploying unix access – options

Saves space
Central System
  • Single Server
    • Single root acc’nt
    • Limited student exp admin tasks
    • No kernel access
  • Virtual Machines
    • Slow or $$
    • Multiple root acc’nt
    • Kernel access

Wastes space
Distributed System
  • Dedicated unix labs
    • Multiple root acc’nt
    • Kernel access
    • Good speed
    • Costs
    • Tech support (reimaging)
Deploying unix access – options

Lots of students needing flexible access

Saves space

Central System

(RULE)

Virtual Host

Single Server

Virtual Machines

Virtual (RULE V2)

Distributed System

Dedicated unix labs

Wastes space

• Single root acc’nt
• Limited student exp admin tasks
• No kernel access

• Slow or $$
• Multiple root acc’nt
• Kernel access

• Good speed
• Multiple root acc’nt
• No kernel access

• Multiple root acc’nt
• Kernel access
• Good speed
• Costs
• Tech support (reimaging)

RULE V2

• Use cheap, small footprint motherboards (Mini-ITX) to provide a “real” (not virtual) PCs

• Assign 1 (or more) PC to each (group of) student(s)

• Retain remote access advantages
  • Use existing Windows labs
  • Access anywhere/anytime
Micro-PCs

- 12V DC
- Low power
- Low heat
- HD or Flash Disk

Multiple “real” Computers

RULE

Switch

VLAN 1

VLAN 2

VLAN 3

Hosts

Terminal

Console

Firewall

ITS

Internet

RULE Server running:
- NFS, TFTP
- and DHCP
- Apache,
- MySQL and
- PHP
Terminal Server

RULE

Switch
VLAN 1
VLAN 2
VLAN 3
Hosts

Terminal Console

RULE Server running:
- NFS, TFTP and DHCP
- Apache, MySQL and PHP

Network Booting

RULE

Switch
VLAN 1
VLAN 2
VLAN 3
Hosts

Terminal Console

RULE Server running:
- NFS, TFTP and DHCP
- Apache, MySQL and PHP
Conclusion

- Teach IP networking fundamentals in unix context
  - Flexible learning & remote access
- Developed & Deployed RULE
  - Richer student experience than a central unix box
  - Easier to deploy & manage than dedicated unix labs
- RULE v2 under development
  - Improved flexibility, requires limited floor space, cheaper than dedicated labs
- Leverage investment in existing Win laboratories