From Microsoft® To Linux® (Relatively) Painlessly

by
Jon "maddog" Hall
Executive Director
Linux International

Trademarks

- Linux is a trademark of Linus Torvalds in several countries

- Unix is a trademark of X/Open in several countries

- Micro$oft would like to own everything else

First Steps

- Learn
- Plan
- Execute
- Contribute
- Evangelize

Learn

- You do not have to learn *everything* at first

- *You* may not have to learn *everything* *ever*:
  - Large Organizations – System Admins
    - Clone desktops
    - Answer questions
  - Smaller Organizations – System Consultants
    - Install systems
    - Provide training
    - Answer Questions
  - Very Small – VARs
    - Turn Key systems

If You Have To Learn:

- Wide variety of learning tools
  - Unix/POSIX
  - Books
  - Courses
  - Magazines
  - HOWTOs
  - LUGs (mailing lists, meetings, helpful people)
- Certifications
  - LPI (www.lpi.org)
  - Red Hat Software, Novell

Plan

- List all programs used today
  - List by application, not by name
  - Understand features used, not features offered
- See if applications have been ported to Linux
  - A lot of major applications have been ported, and others are planned
- See if acceptable alternates exist
  - www.freshmeat.org
  - www.slashdot.net

www.sourceforge.net

- Investigate what is there
  - 97,000+ projects
  - 1,000,000 + registered developers
- Set up small test system to test functionality

Planning Strategies

Marketing teaches you:

- To eat like a hog

- To drink like a fish

- To smoke like a chimney

- To have sex every hour

- For heaven's sake, don't eat refined sugar!

Planning Strategies:
Go and Sin no more!

- Use Free and Open Source Software for new projects
  - Portable languages (Perl, Python, JAVA, LISP, C)
- Use FOSS on top of proprietary operating systems
  - Open Office
  - Standard-supporting web browser (and produce standard pages)
  - MySQL/Postgres
  - Dual OS products

Planning Strategies: Low Hanging Fruit

- Replace redundant systems
  - DNS Servers
  - Firewalls

- Replace expensive hardware:
  - Database engines

- Replace "invisible" functionality
  - File and print servers (SMB, NFS, Appletalk, Novell)

Investigate Groupware Alternatives

- Evolution

- SuSE's Exchange replacement

- Software Libre projects
  - gnomemeeting
  - VoIP
  - Whiteboard

Investigate Software Libre Mega Projects

- ERP
- CRM
- Accounting
- Project Management

Some are simple, but improving rapidly and may meet your needs

Investigate Custom-built Software

Built using GPLed software to your specifications

- Databases
- OpenGIS
- GNUplot
- Scripting Languages
- Printing and viewing libraries

Investigate Restructuring Work

- Do your employees ALL need a full desktop?

  - Linux Terminal Server Project - www.ltsp.org

- Do your employees/students need to use MS Office?

  - Do they communicate with outside customers?

  - Do they use externally created documents?

  - Influence document suppliers to use standards

  - Do employees HAVE to play "mines" and "solitaire"?

  - Can you reduce the number of systems running proprietary software?

Execute

- Cold Turkey
  - Convert after pilot project
- New project co-existence
- Gradual replacement over time
  - For stubborn applications use:
    - Dual Boot
    - VMware
    - Win4Lin
    - Crossover Office (Codeweavers)

Execution Tips:

- Find "enthusiastic" end user
  - If none, make an end user enthusiastic
- Make sure old backup data still available
- Use hardware emulators for legacy applications
- umsdos and FreeDOS
- User Mode Linux for testing new versions of Linux with applications

Execute "Do Nots"

- Do not convert good-performing, stable project
  - All pain, no gain
  - Exception:
    - Expensive hardware
    - Invisible to end users (SPARC Oracle server)
- Do not convert project without successful pilot
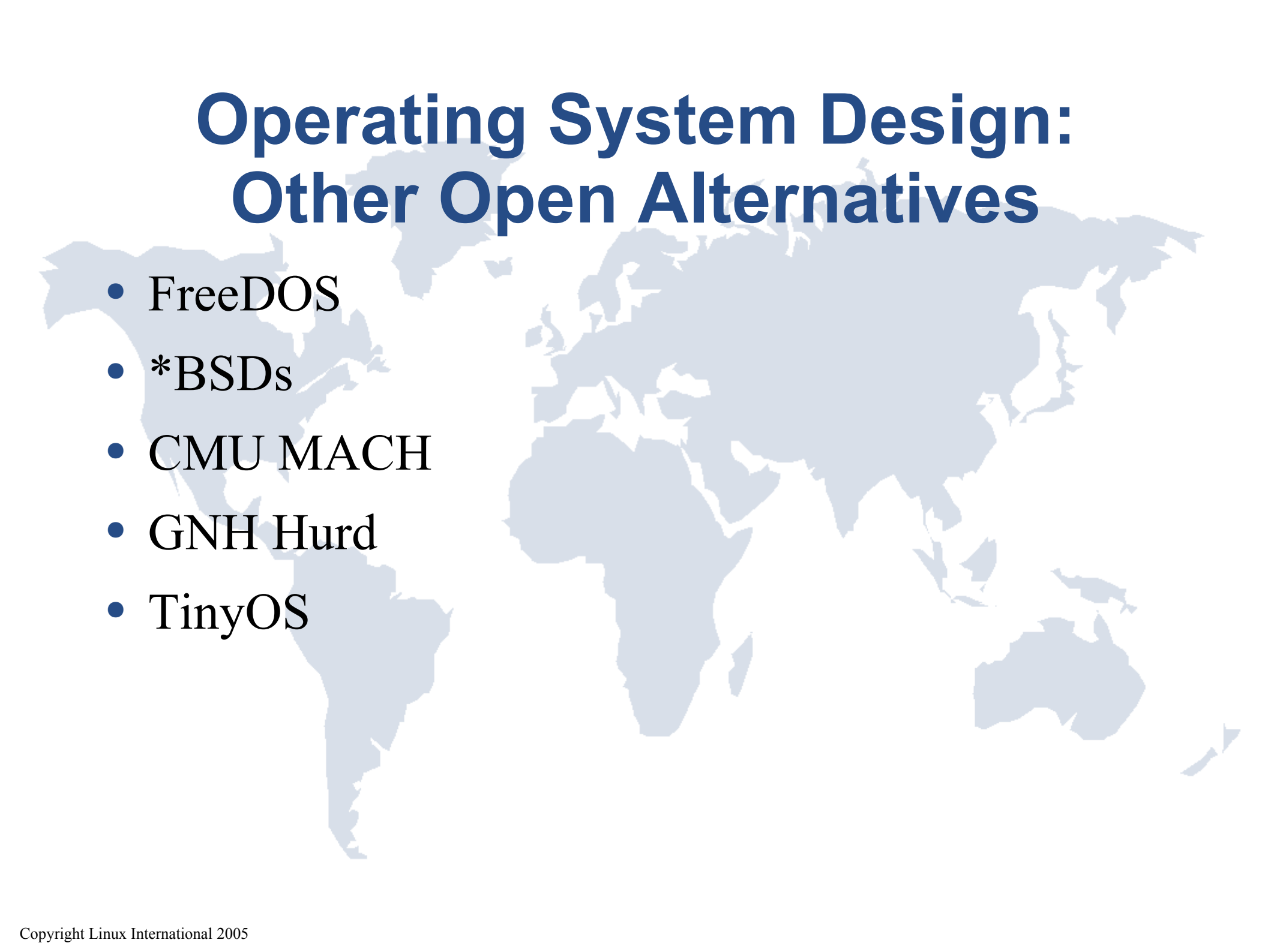
Teaching Computer Science

- Use GNU compiler suite
- Use MS Unix compatibility tools to teach shell scripting
- User portable languages (not VB, VC)
- Use mono, not .NET
- Use Delphi/Kylix
- Use Linux
  - 1/3 of all servers, most supercomputers, embedded

Operating System Design: Linux

- monolithic kernel
- loadable kernel modules
- loadable device drivers
- multi-architecture
- multi-CPU
- multi-user
- 32 and 64 bit

Operating System Design: Other Open Alternatives

- FreeDOS

- *BSDs

- CMU MACH

- GNH Hurd

- TinyOS

Database Design

- PostgresSQL
  - Relational and object oriented
- MySQL

Other Schools

- Engineering/Science – >3500 applications
- Humanities – Project Guttenburg
- Business – Open Source projects
- K-12 teaching
- MIT

Students can help improve existing projects

Administrative

- SAGU
    - scheduling
    - reports
- Library programs

Introducing Students

- University laptop programs
  - New machines, end of production
  - Sold/leased to students
  - FOSS installed
- CDs of FOSS courseware for incoming students
  - Windows
  - Linux
- Professors have proprietary and FOSS

Contribute

- GPL your own code when possible
- Contribute to the community
  - hardware
  - documentation
  - money
- Hire FOSS developers

Evangelize

- Tell others about your project
  - Write articles
  - Talk at conferences
  - Talk to business groups
- State the good and the bad

Have Fun!

Questions?