

Free and Open Source In Education: Why FOSS Teaches Students Twice

by
Jon "maddog" Hall
Executive Director
Linux[®] International[®]

Trademarks



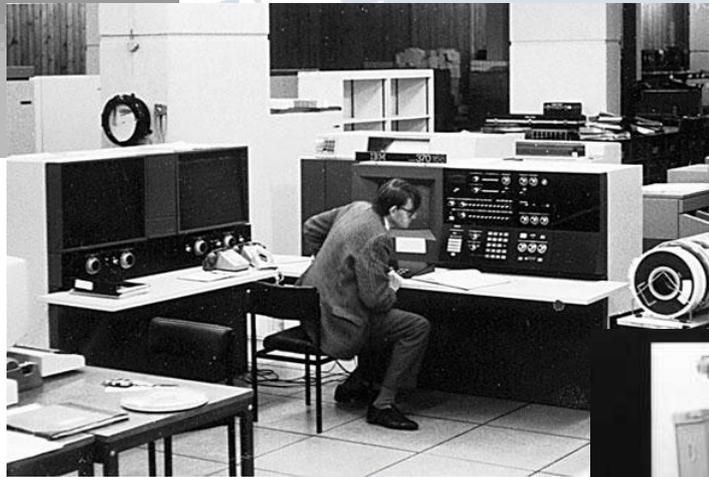
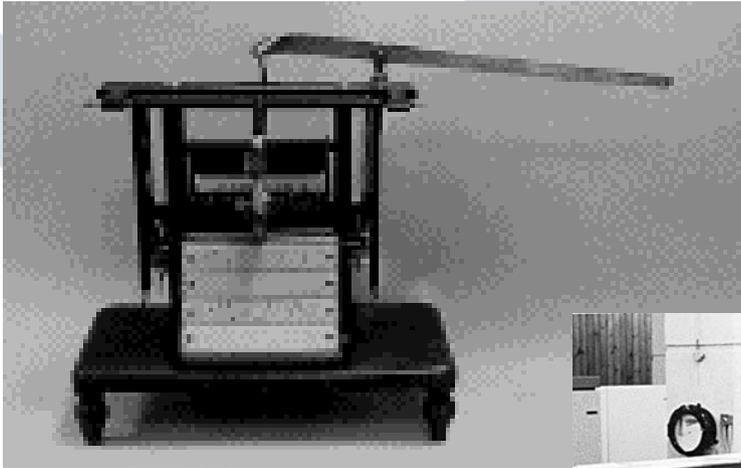
- Linux is a registered trademark of Linus Torvalds in several countries
- Unix is a trademark of X/Open in several countries

Software Li[bv]re!

- Free as in *freedom*, not free as in *beer*.
 - Free to read the source code
 - Free to make changes to the source code
 - Free to redistribute those changes
 - *Free to use the software for any purpose*
 - You can not limit another's freedoms

Richard Stallman has *never* said you should not make money writing software.

A Bit of History



Once “All” Software was “Free and Open Source”

- Software written to specification
 - Inputs, outputs and steps to be taken were specified
 - contract was written
- Criteria for acceptance
 - Bug rate
 - Documentation
 - On time
 - Other...



Software Owned By Customer

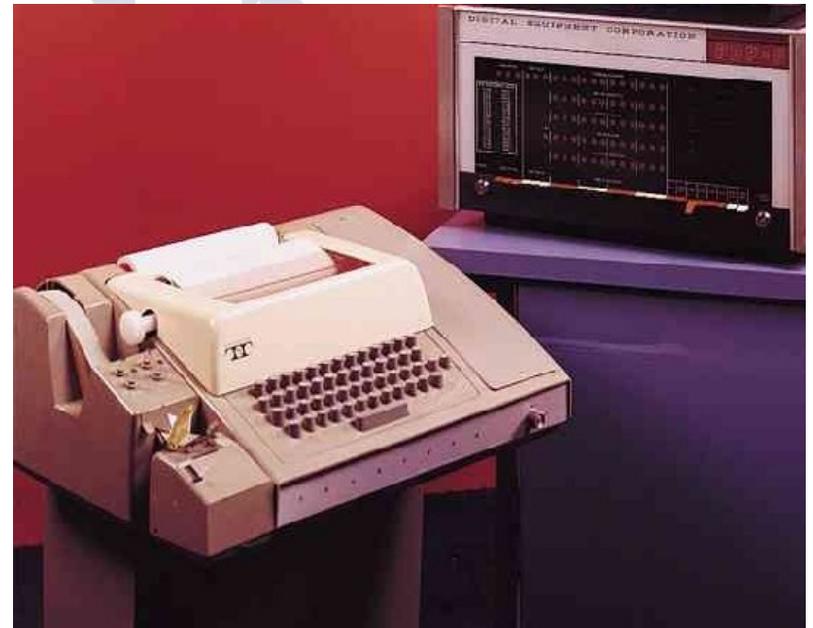
If changes needed, could select vendor

- Bugs fixed
- Extensions

Some Software Was Even “Free”

DECUS

- A Text Editor or....
- Why did they do this?
- Amateurs vs Professionals
 - Amateur painters
 - Amateur athletes



Some Problems Did Exist

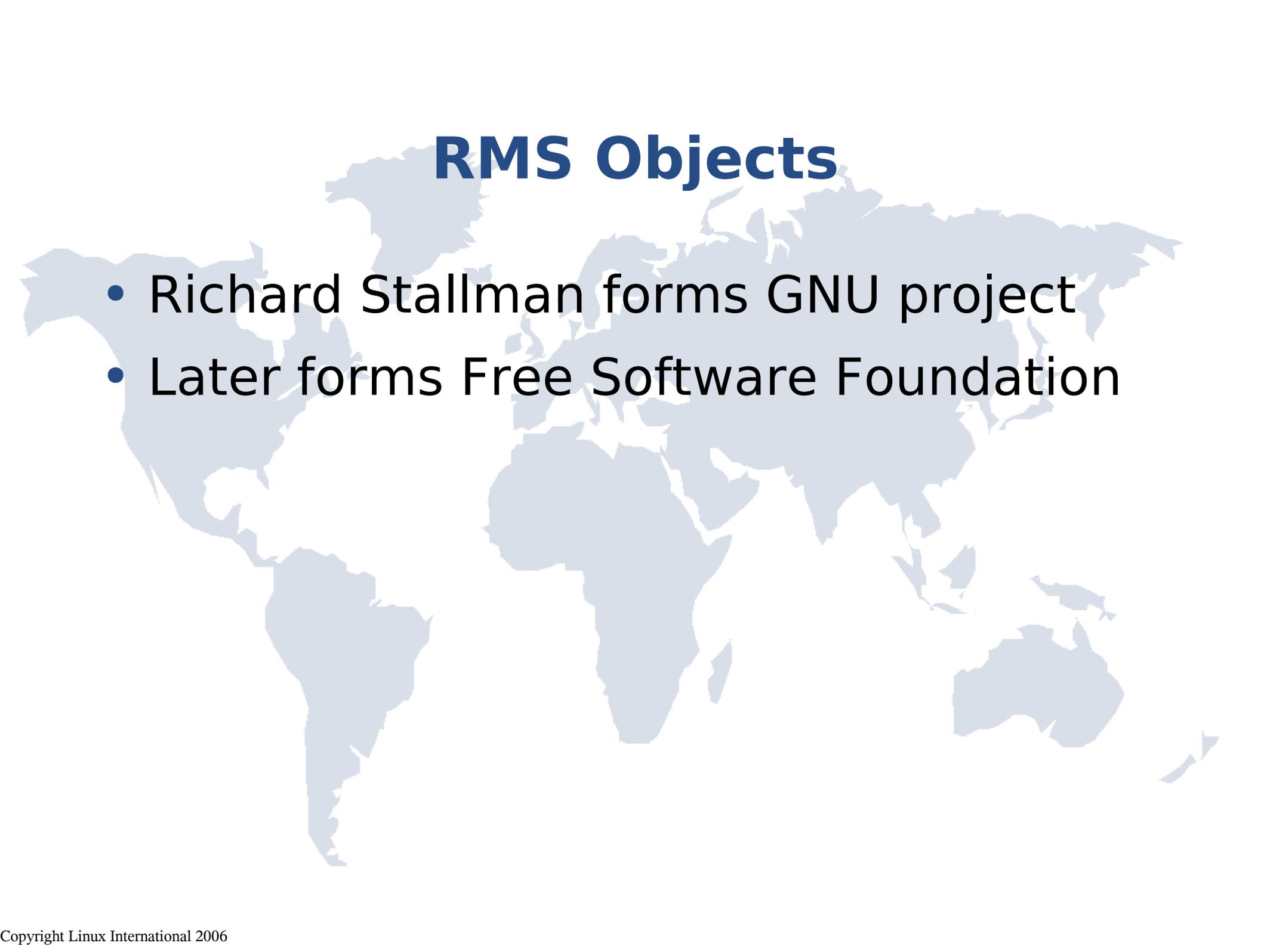
- Too expensive to give a student a computer
- Communication was by snail mail and uucp
 - “You are going to email me HOW MUCH?”

The Death of Middle Earth

1980-1983

- Commercialization of Unix
 - Sun Microsystems – 1981
- PC happens
 - Low price hardware and software
 - Microsoft, Apple and others
 - Growth of the Computer Store

RMS Objects



- Richard Stallman forms GNU project
- Later forms Free Software Foundation

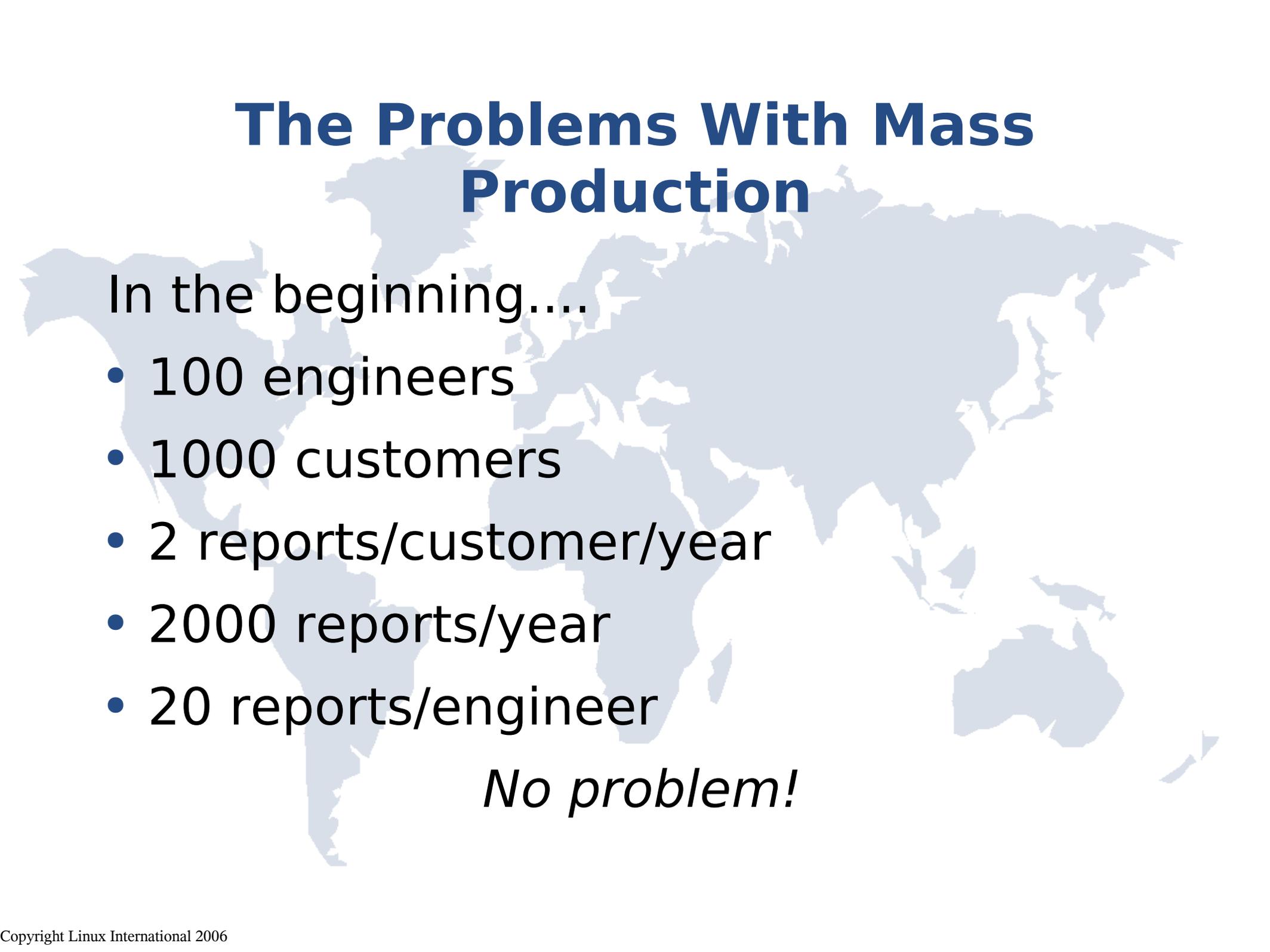
Vendors Respond To Universities



- University Discount
- Student editions
- Universities turn to Microsoft
 - At first, in non-CS courses
 - Later, all of university
- “Lock them in when they are young”

“You can't make any money from a University”

The Problems With Mass Production



In the beginning....

- 100 engineers
- 1000 customers
- 2 reports/customer/year
- 2000 reports/year
- 20 reports/engineer

No problem!

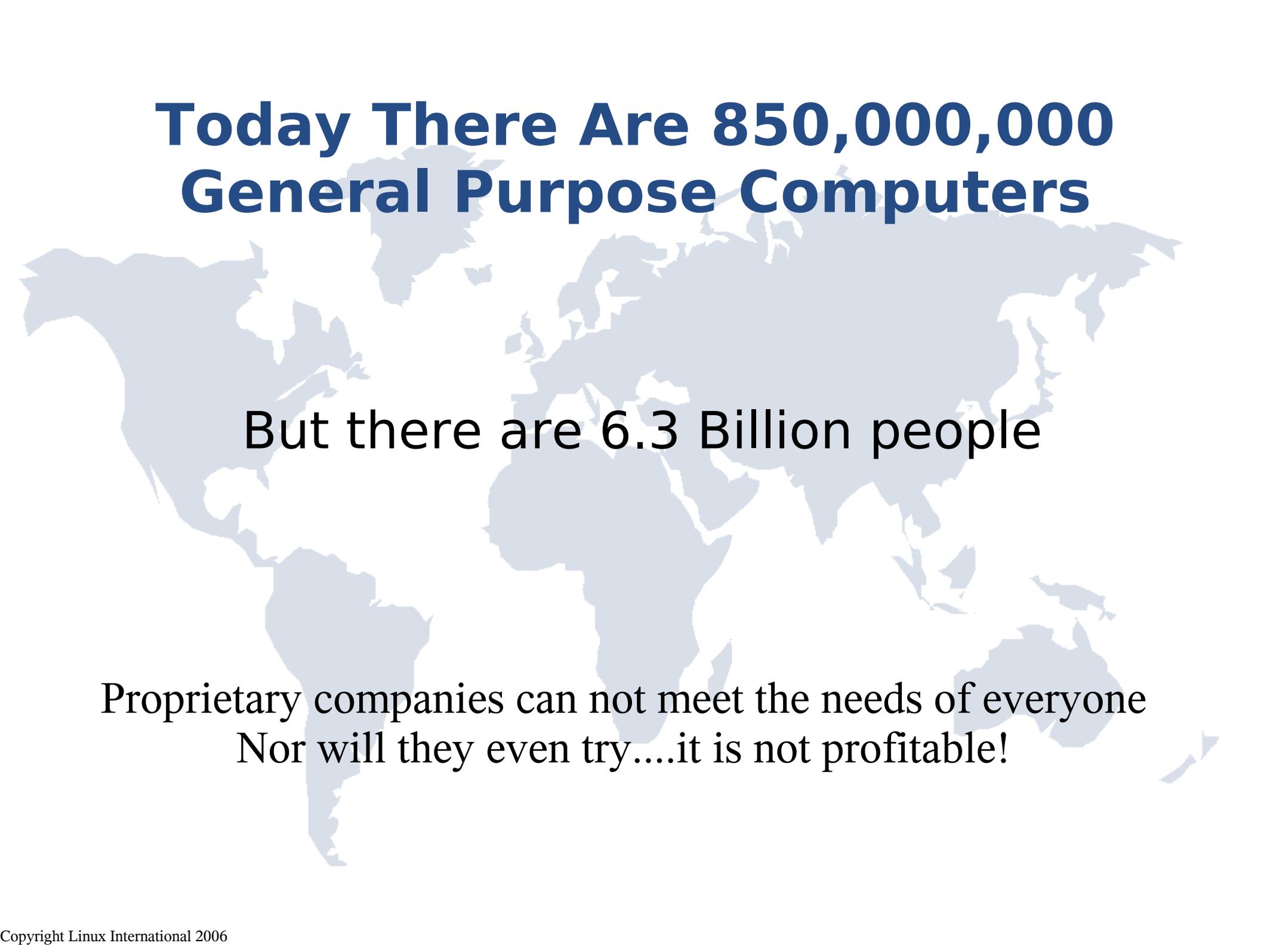


Today...

Our company has grown

- 200 engineers
- 4.5 million customers
- 2 reports/customer/year...

Do you see the problem?



Today There Are 850,000,000 General Purpose Computers

But there are 6.3 Billion people

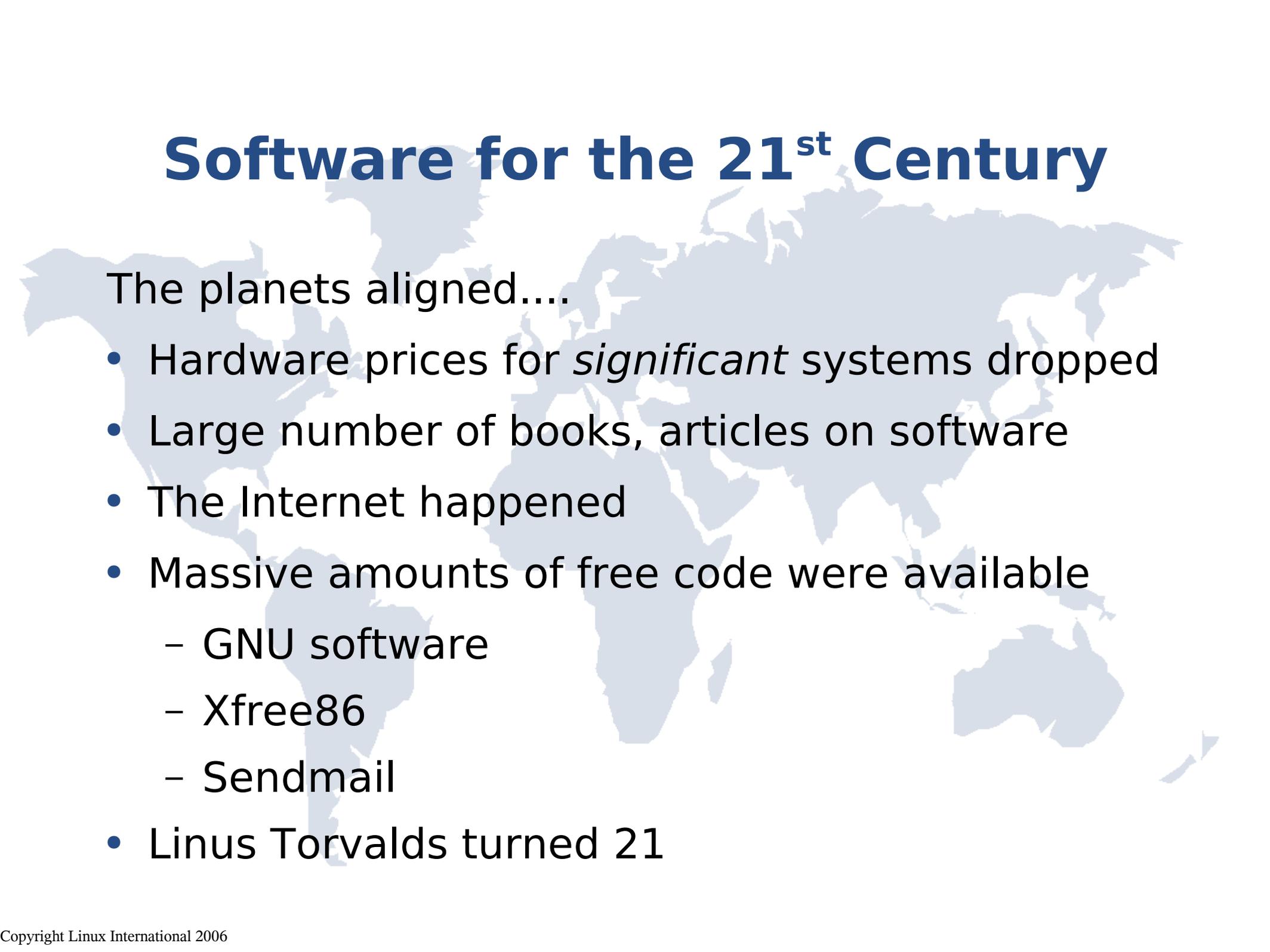
Proprietary companies can not meet the needs of everyone
Nor will they even try....it is not profitable!

Binary-Only Software

What Else Is Wrong?

- Can not see how software works
- Can not learn from existing software
- Hard to do research
 - Hard to exchange research efforts
 - Hard (or impossible) to get research out to consumers

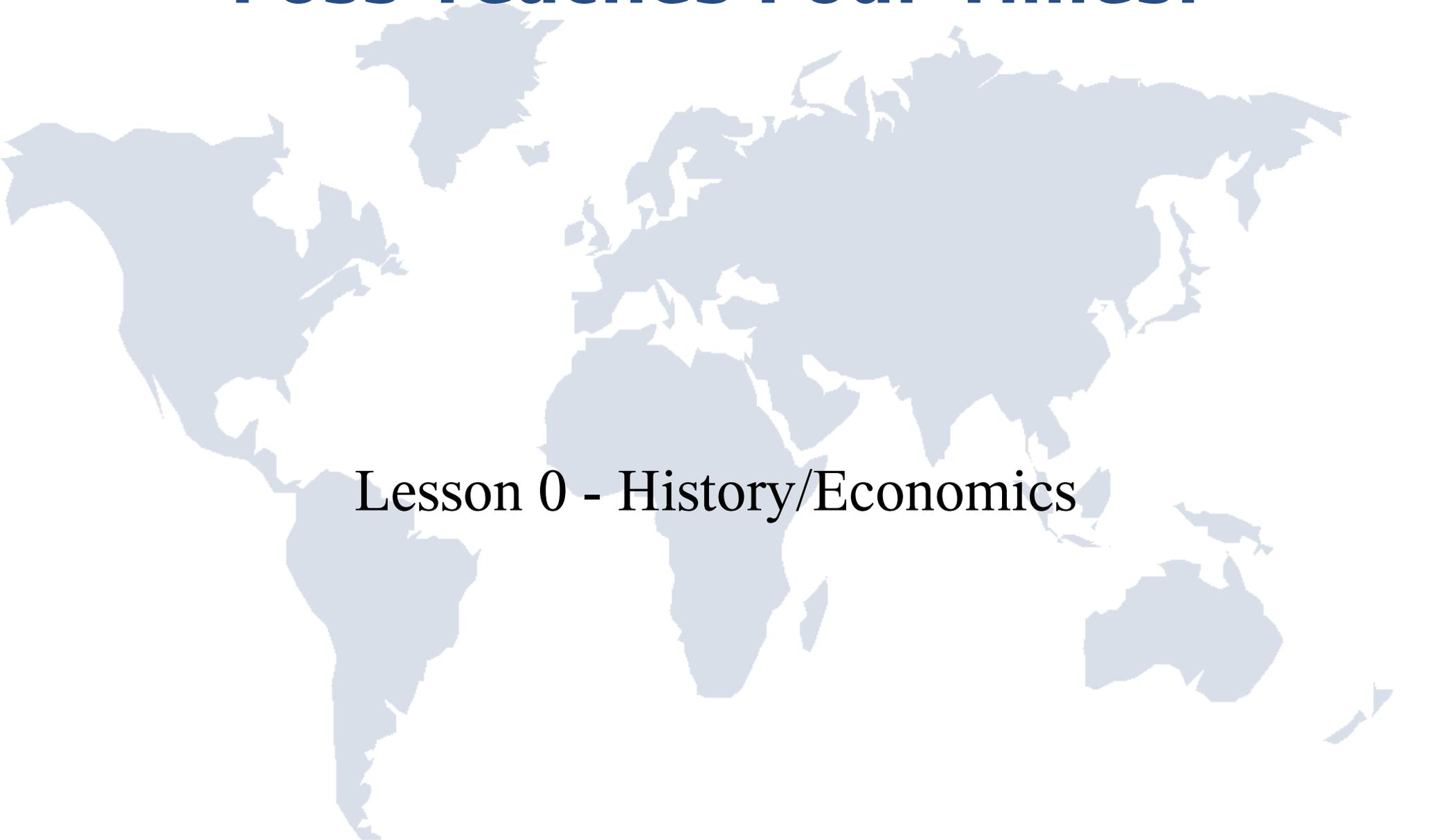
Software for the 21st Century



The planets aligned....

- Hardware prices for *significant* systems dropped
- Large number of books, articles on software
- The Internet happened
- Massive amounts of free code were available
 - GNU software
 - Xfree86
 - Sendmail
- Linus Torvalds turned 21

Foss Teaches Four Times:



Lesson 0 - History/Economics

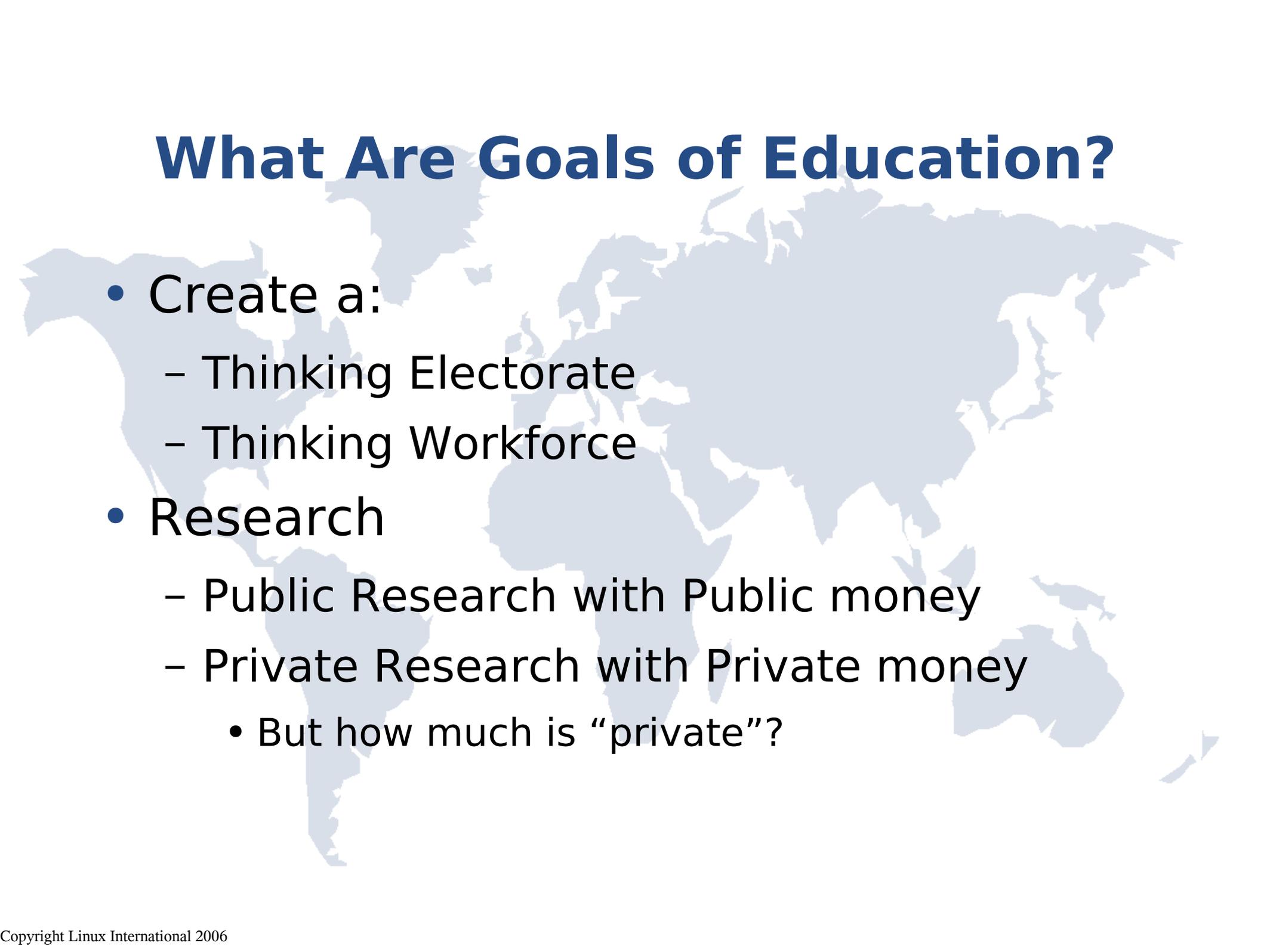
FOSS Teaches You Four Times:

**Lesson I - All software “teaches”
you how to use it.**

What Are Goals of Education?

To get a good paying job!

What Are Goals of Education?



- Create a:
 - Thinking Electorate
 - Thinking Workforce
- Research
 - Public Research with Public money
 - Private Research with Private money
 - But how much is “private”?

FOSS Teaches You Four Times:

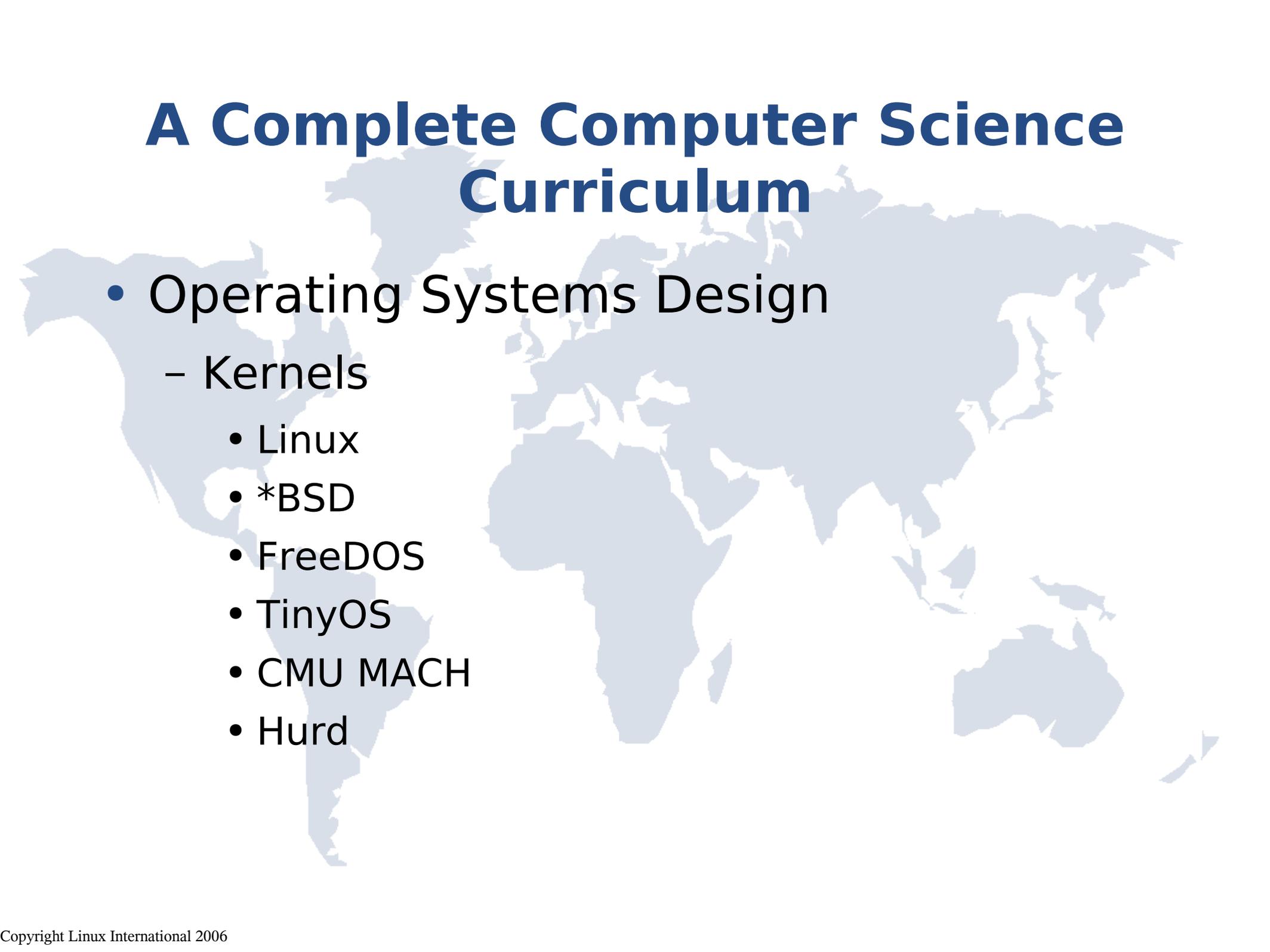
**Lesson II – How does software do its
job?**

Importance To Country's Economy



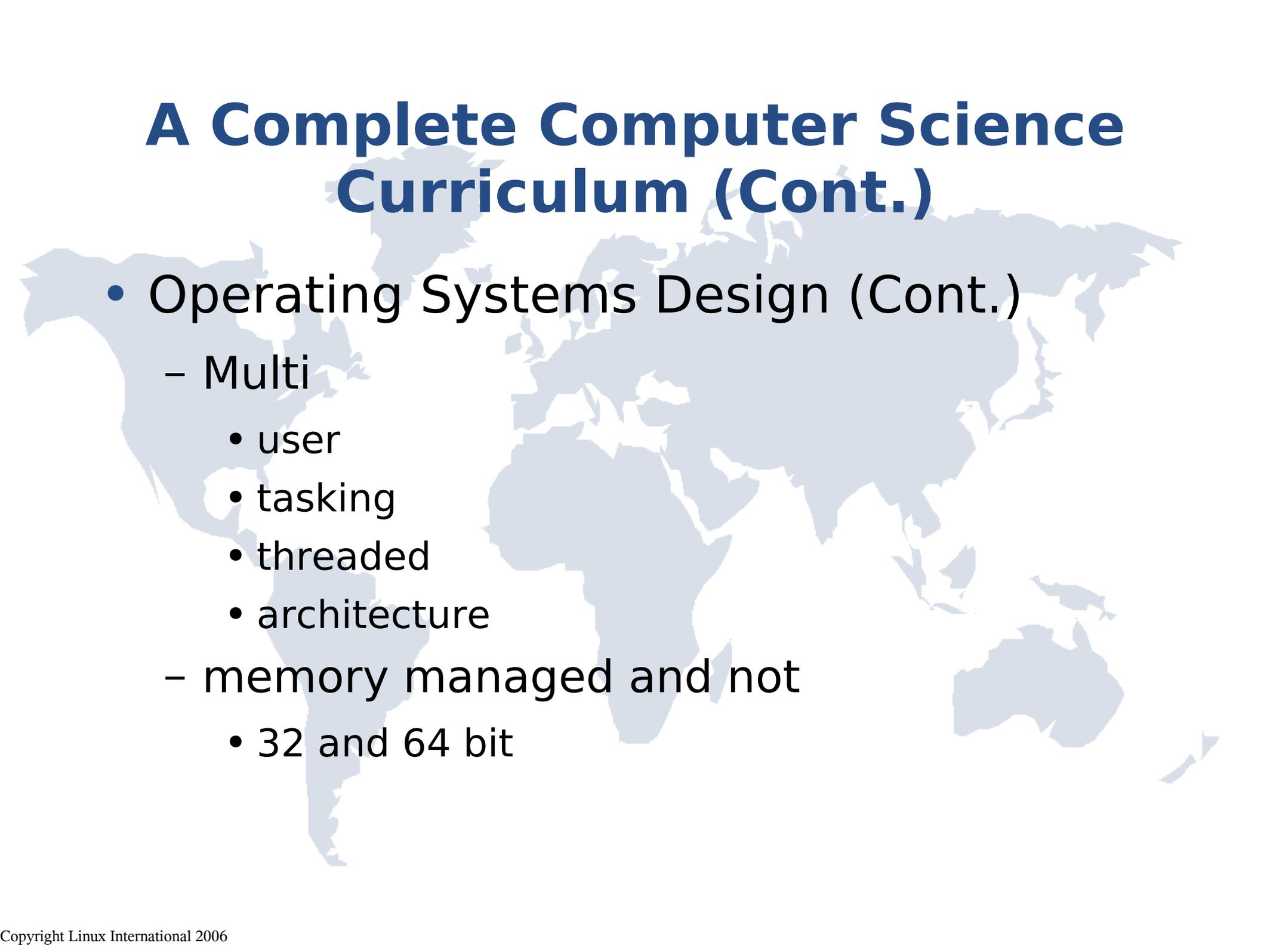
- Local jobs
 - High Tech
- Balance of Trade
- Security of country
 - Economic
 - Military
 - Longevity
- Eliminate piracy
 - Ethics

A Complete Computer Science Curriculum



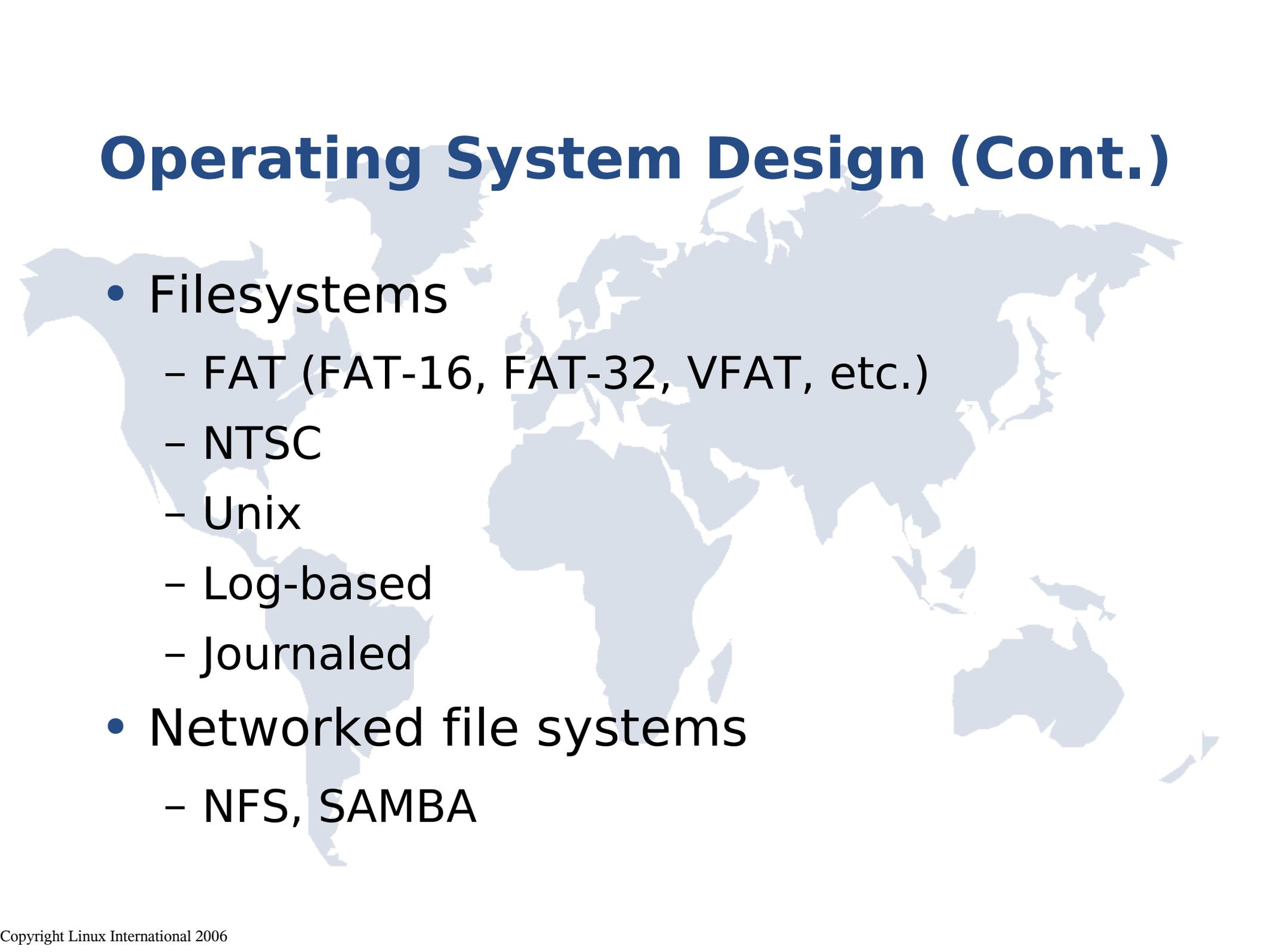
- Operating Systems Design
 - Kernels
 - Linux
 - *BSD
 - FreeDOS
 - TinyOS
 - CMU MACH
 - Hurd

A Complete Computer Science Curriculum (Cont.)



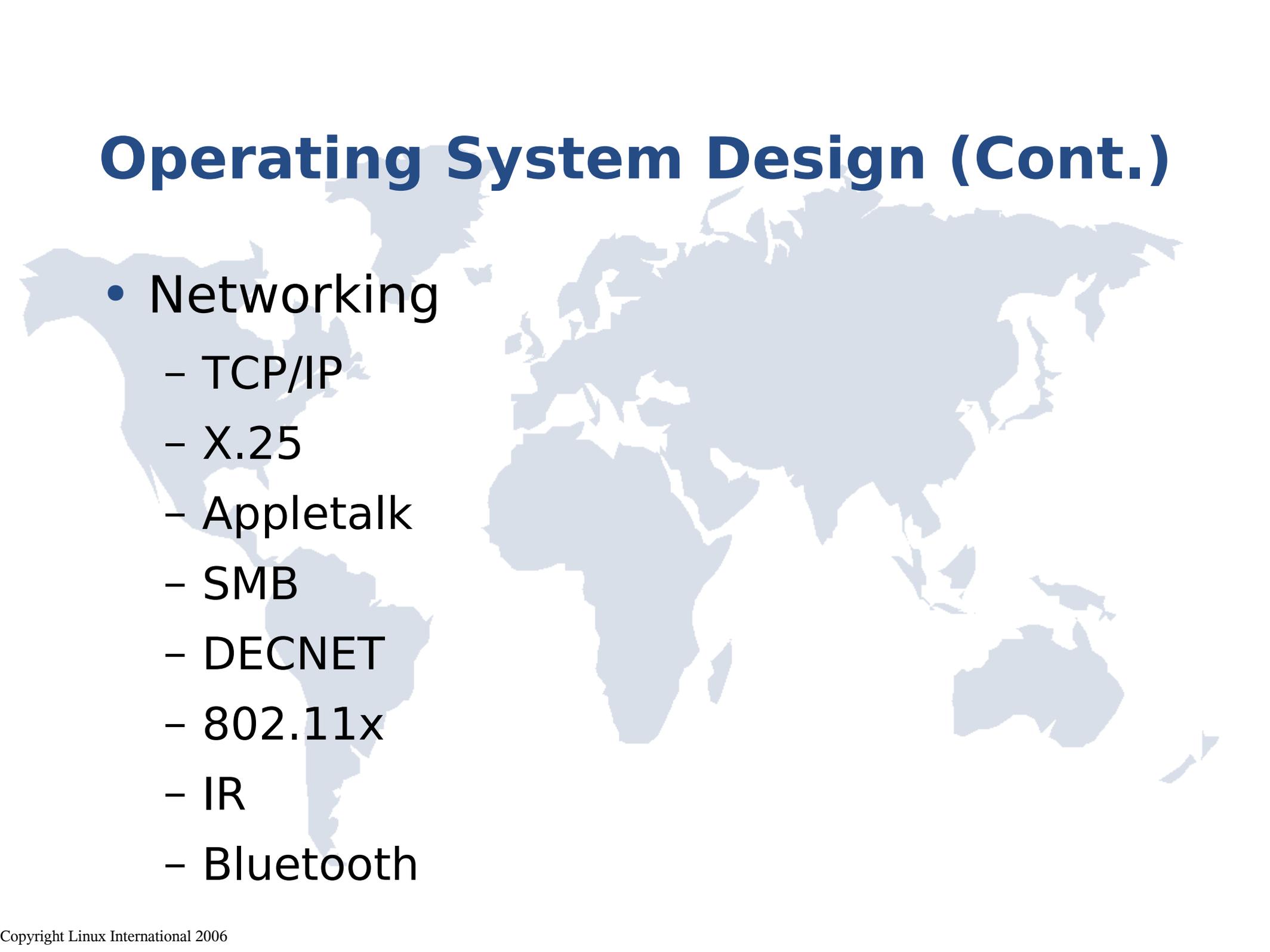
- Operating Systems Design (Cont.)
 - Multi
 - user
 - tasking
 - threaded
 - architecture
 - memory managed and not
 - 32 and 64 bit

Operating System Design (Cont.)



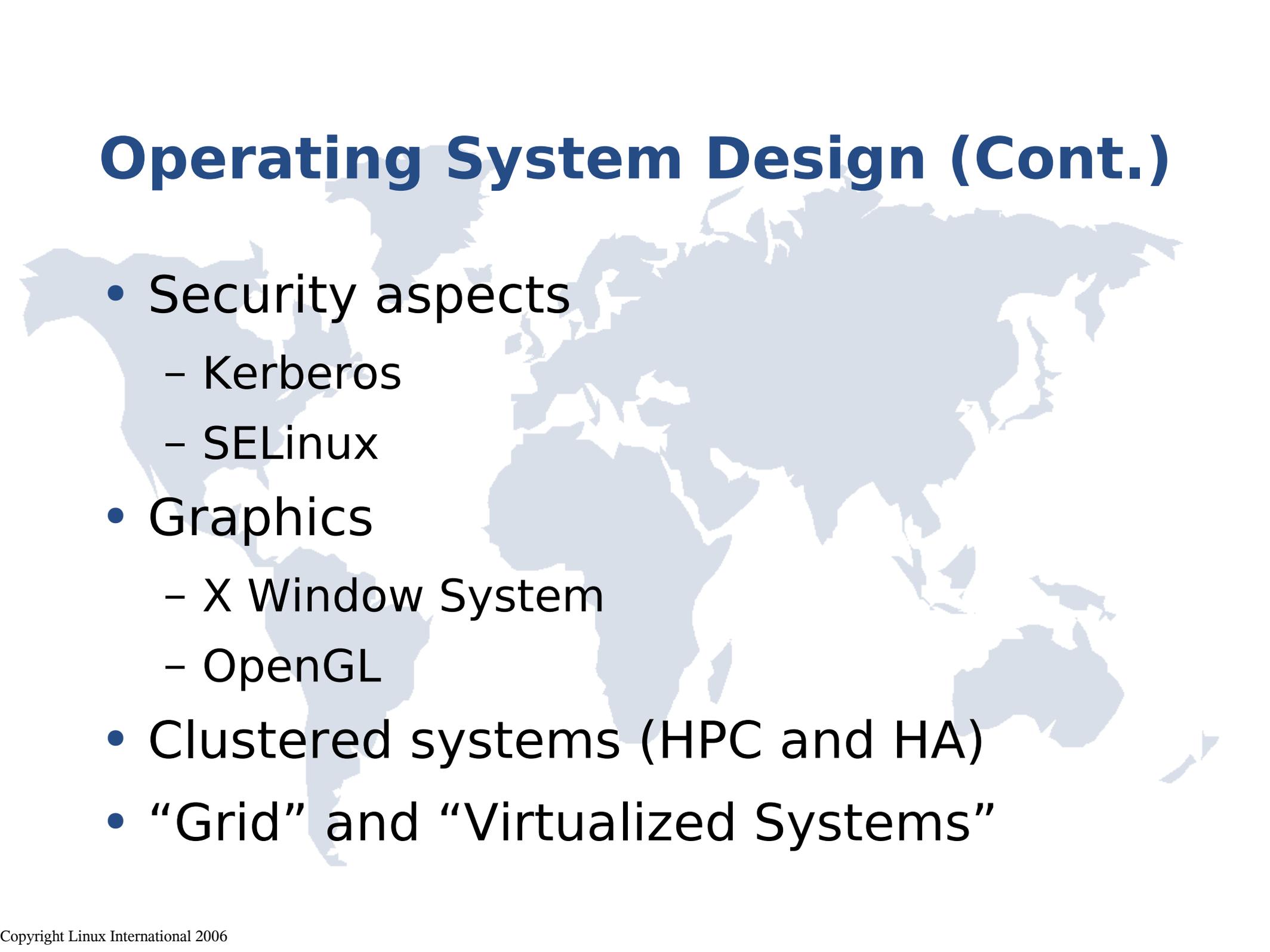
- Filesystems
 - FAT (FAT-16, FAT-32, VFAT, etc.)
 - NTFS
 - Unix
 - Log-based
 - Journalled
- Networked file systems
 - NFS, SAMBA

Operating System Design (Cont.)



- Networking
 - TCP/IP
 - X.25
 - Appletalk
 - SMB
 - DECNET
 - 802.11x
 - IR
 - Bluetooth

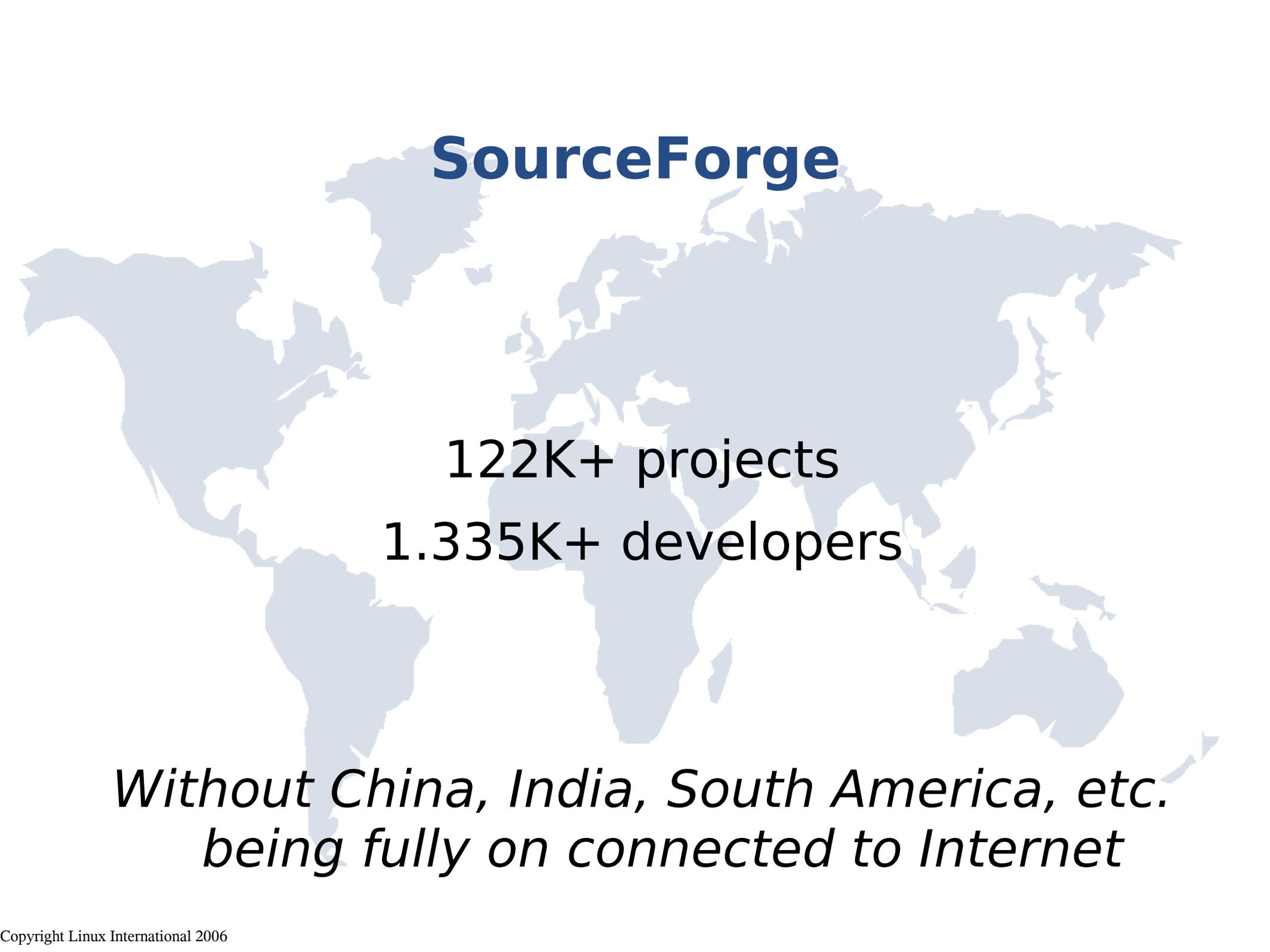
Operating System Design (Cont.)



- Security aspects
 - Kerberos
 - SELinux
- Graphics
 - X Window System
 - OpenGL
- Clustered systems (HPC and HA)
- “Grid” and “Virtualized Systems”

Not Just “An Operating System”

- Compilers
 - “C”, C++, Fortran, Pascal, Lisp, BASIC, etc.
- Interpreters
 - Python, Perl, Ruby, Tcl/Tk
- Database engines
- Office Systems
- Multimedia tools
- VoIP



SourceForge

122K+ projects

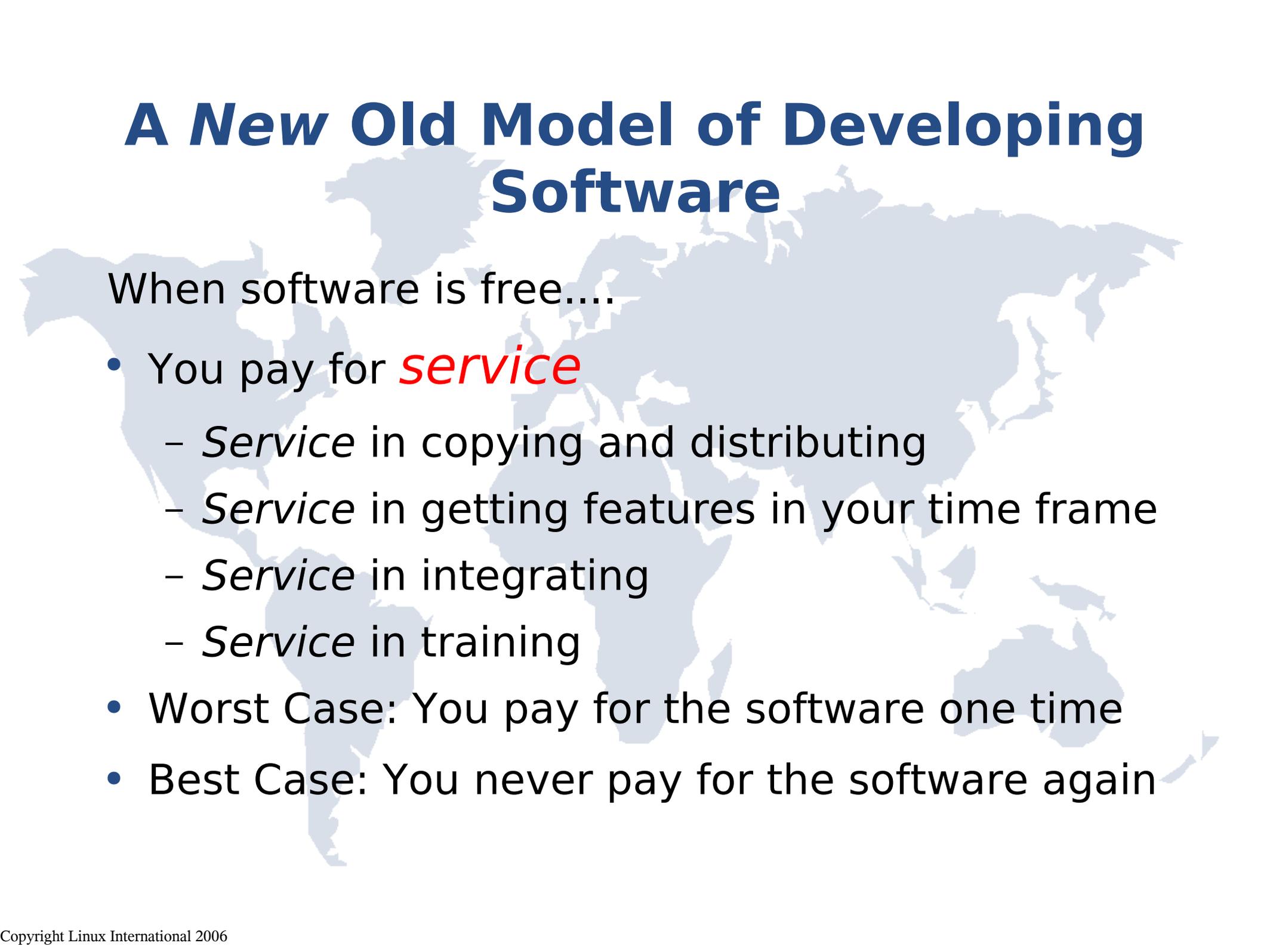
1.335K+ developers

*Without China, India, South America, etc.
being fully on connected to Internet*

FOSS Teaches You Four Times

Lesson III - Cooperation and Collaboration

A *New Old* Model of Developing Software



When software is free....

- You pay for *service*
 - *Service* in copying and distributing
 - *Service* in getting features in your time frame
 - *Service* in integrating
 - *Service* in training
- Worst Case: You pay for the software one time
- Best Case: You never pay for the software again

Do Not Be Afraid Of The Word *Service*

- More than “just installing software”
- This is Brain Surgery
- This is like being a lawyer

Things to Teach In New Model

- How to do distributed development
- How to license software
- How to develop formal standards
- How to write code to standards
- How to motivate software developers
- How to locate and engage the community of users and developers
- How to innovate, everywhere, always

How to evaluate and size customer needs and desires

More Things To Teach (and not teach)

- Teach:
 - Fundamentals
 - How does computer really work?
 - Machine language
 - Cache
 - How do compilers, OS really work?
 - Comparison evaluation
 - Various office packages
 - How to share
- Do not teach:
 - Specific products

Fiji - 1995

- Returning from DECUS Australia
- Stopped off for “rest” in Nadi
- University of Pacific

I Felt A Bit Like.....



Curitiba, Brazil

- High School that had everything
- High School that had “nothing” ...
 - Except pride



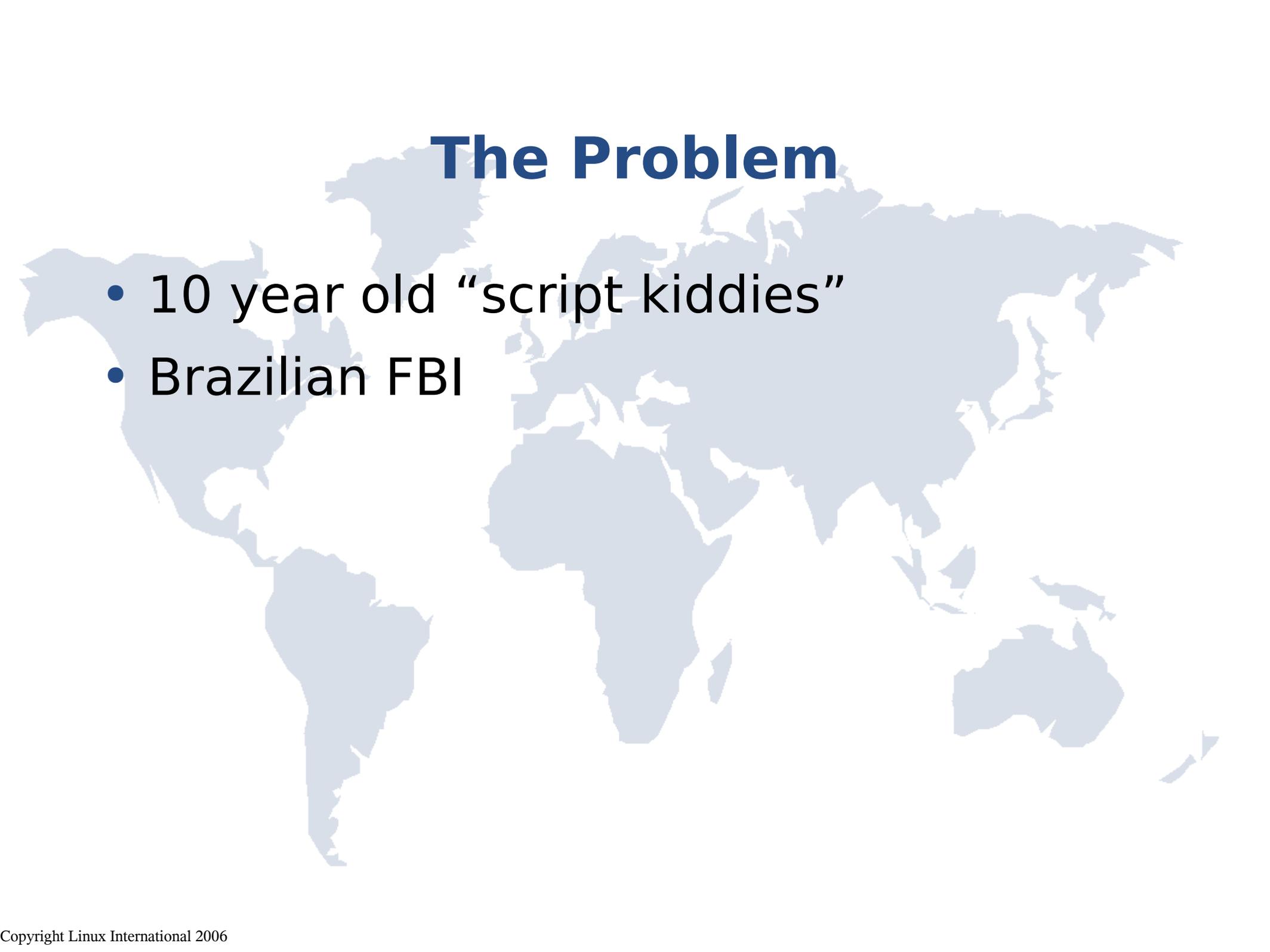
Charter Schools

“North East Linux Symposia”

- Held at private school (good food!)
- Held during early and late summer
- All teachers (well, almost all)
- Hands-on training
- Share experiences, successes
- David Trask
 - dtrask@vcsvikings.org
- Matt Oquist
 - moquist@majen.net



The Problem



- 10 year old “script kiddies”
- Brazilian FBI

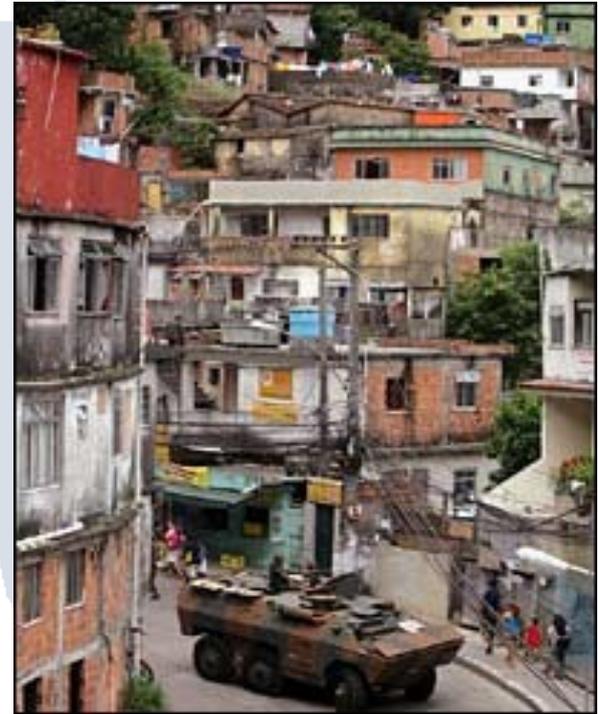
One Solution: Hackerteen



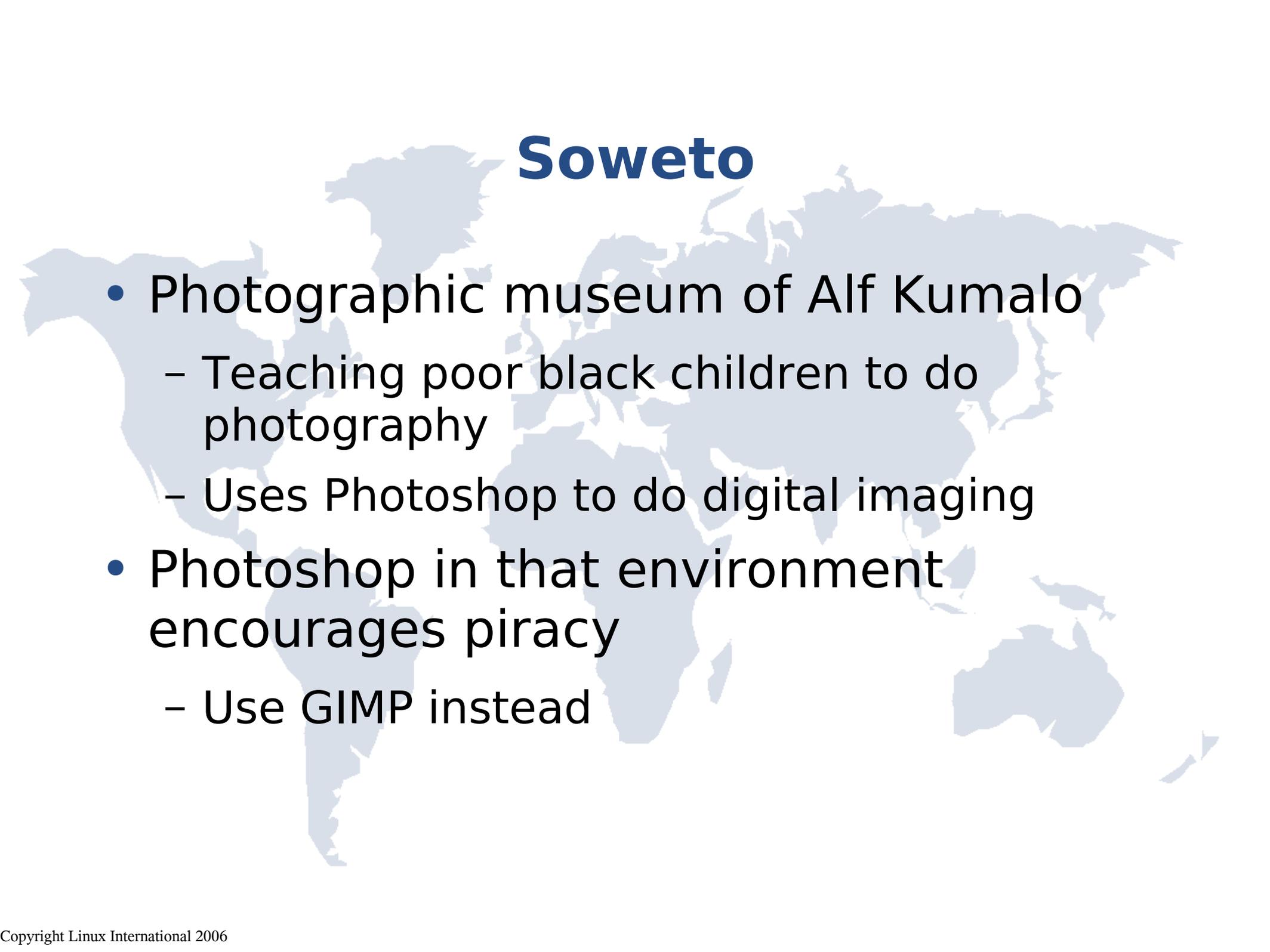
- Create hackers, not crackers
 - Teach them real skills
 - Create pride, self confidence
 - Employ them
- Computer based training
 - Using Japanese-style cartoons to create plot
 - Psychologist and group work once/month
 - Karate-style of belt movement

Results

- Hackerteen graduates now employed
- Hackerteen interns teaching at favelas and Telecentros
- Hackerteen is “for-profit” business
 - scholarships available through charitable donations and government



Soweto



- Photographic museum of Alf Kumalo
 - Teaching poor black children to do photography
 - Uses Photoshop to do digital imaging
- Photoshop in that environment encourages piracy
 - Use GIMP instead

Soweto: The Aftermath

- Opening of FOSS Research Center
- Installation of Wireless mesh over Soweto



More Than Just Software



- Free Standards Group
 - www.freestandards.org
- Linux Professional Institute
 - www.lpi.org
- Linux Training
 - www.lintraining.com
- Open Hardware
 - Simputer (www.simputer.org)
 - SolarPC (www.solarpc.com)
 - Open Telephony

Supercomputers.....

- 1994 – a problem
- Beowulf Systems
 - GNU/Linux
 - PVM, MPI, OpenMP
- Numa machines
 - The same APIs

“The power of an IBM SP/2 for
1/40th of the price.”

- Pat Goda, Los Alamos Labs



...to Workstation Farms....

- Lots of COTS
- Tied together with high-speed networking

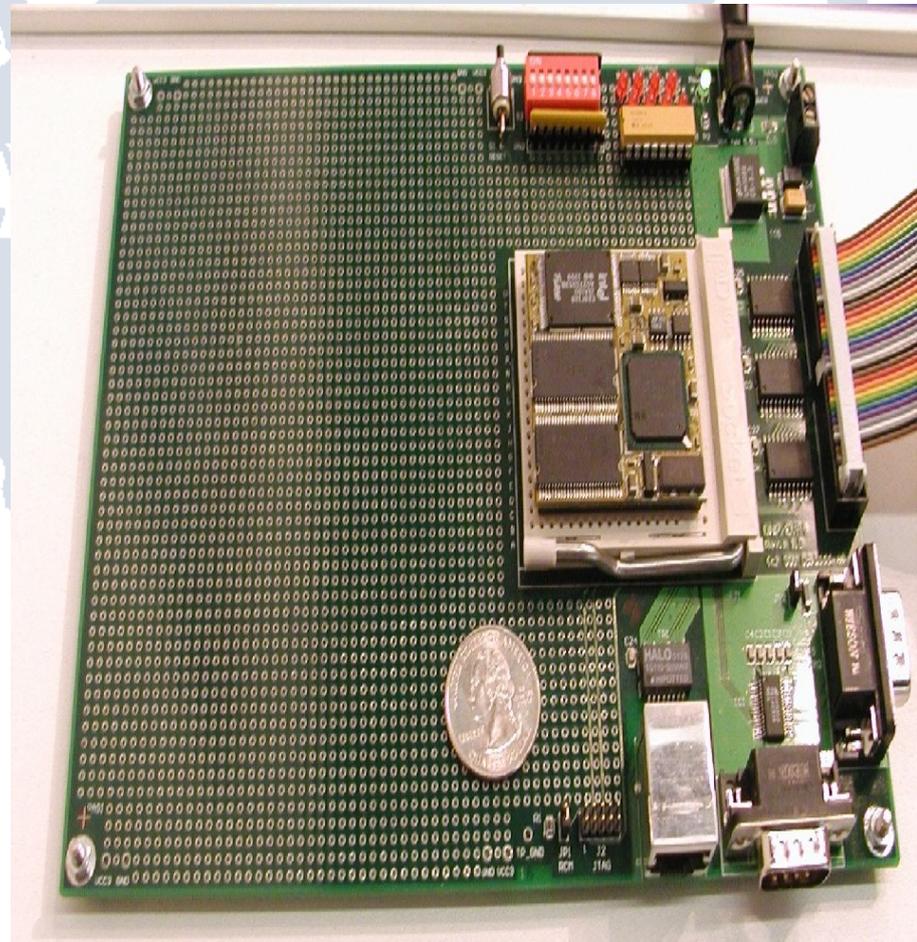


...to “Supercomputing” on a Single PC

- User Mode Linux

....to Embedded Systems....

- Modular kernel
- Multi
 - Architecture
 - User
 - Tasking
 - Secure
 - Stable
- Royalty Free



....Other Embedded Systems...

- Imagine students building products with these
- Imagine students designing these



IBM/Citizen Watch WatchPad

A Challenge for Your Schools

- Find your brightest students
- Get them to create a proposal for an embedded system products
- Choose best five proposals
- Get CS students to develop software on GNU/Linux systems
- Get EE students to develop controllers
- Get companies to manufacture products, create jobs

An Example: Wearable Computers



Some Great Projects



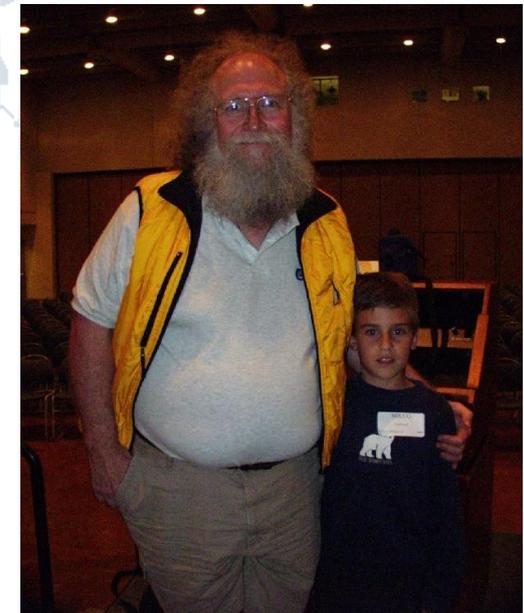
- www.misterhouse.net
- www.asteriskathome.sourceforge.net
- Rosegarden and other audio/video
- www.mythtv

FOSS Teaches You Four Times

Lesson IV – You can never stop learning

Does Closed Source Allow You To Do This?

- A 19 year old president
- A 15 year old kernel developer
- A 14 year old distribution developer
- A 13 year old systems admin
- An 11 year old programmer of supercomputers



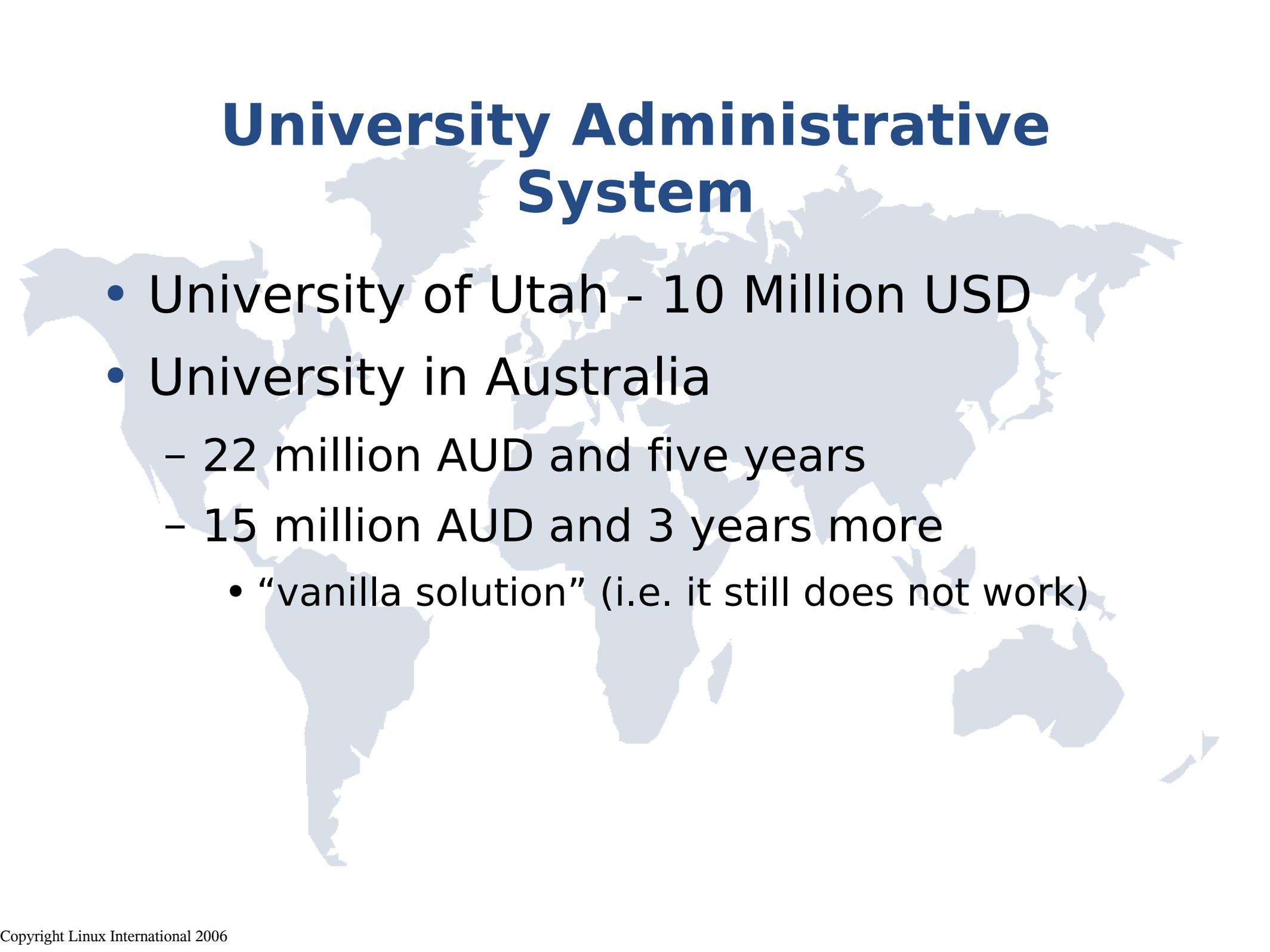
What About Other Disciplines?

- Department of Biological Science working with Department of Computer Science?
- Insurance companies using FOSS techniques?

Let's tear down the walls



University Administrative System



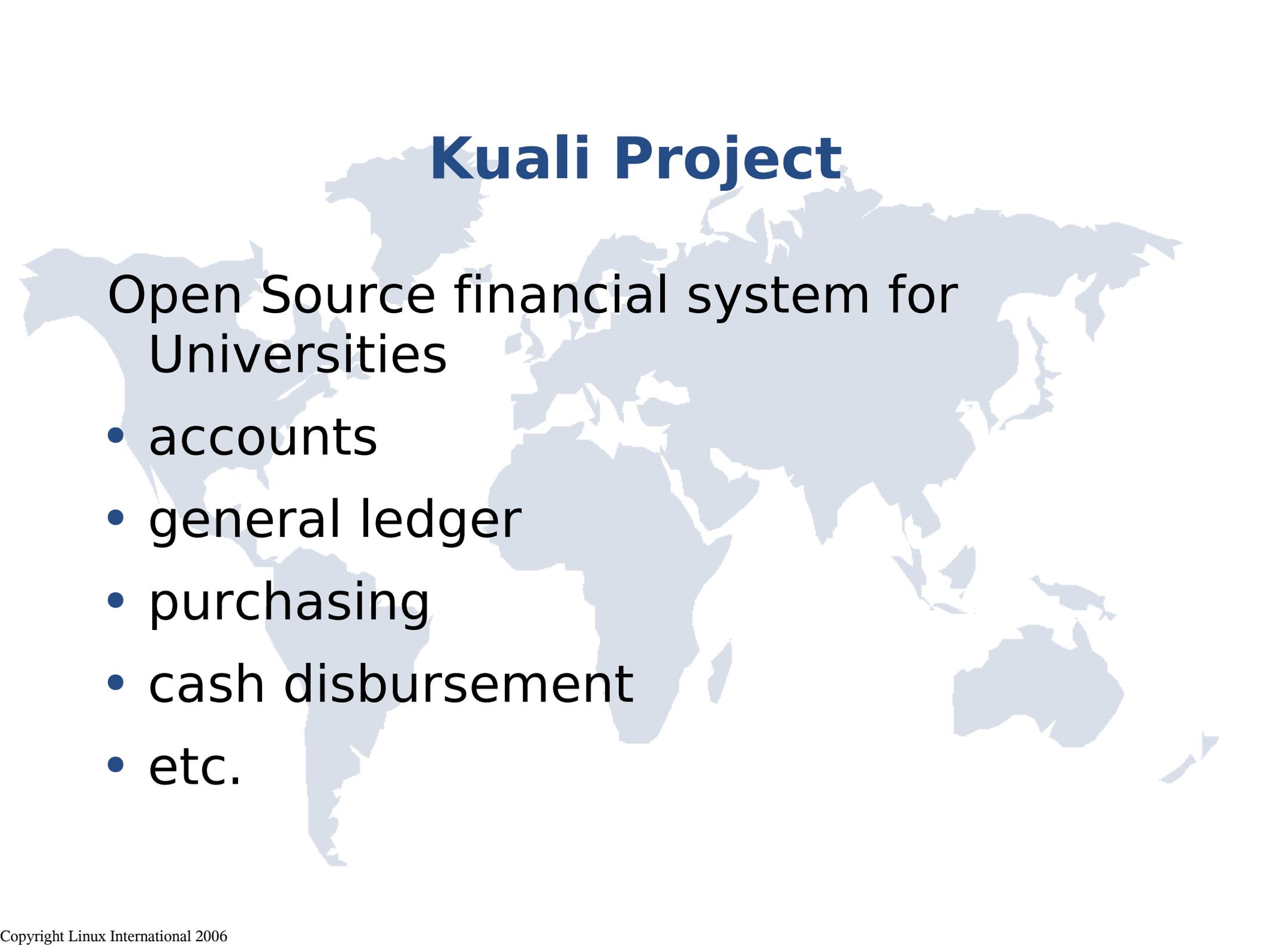
- University of Utah - 10 Million USD
- University in Australia
 - 22 million AUD and five years
 - 15 million AUD and 3 years more
 - “vanilla solution” (i.e. it still does not work)

SAGU

<http://sagu2.solis.coop.br/>

- SAGU – Open Source and GPL
 - Free to universities
 - Incrementally improved
- Entrance Exams and Placement
- Scheduling of teachers, rooms, students
- Financial and Accounting System
- Human Resources and Reporting

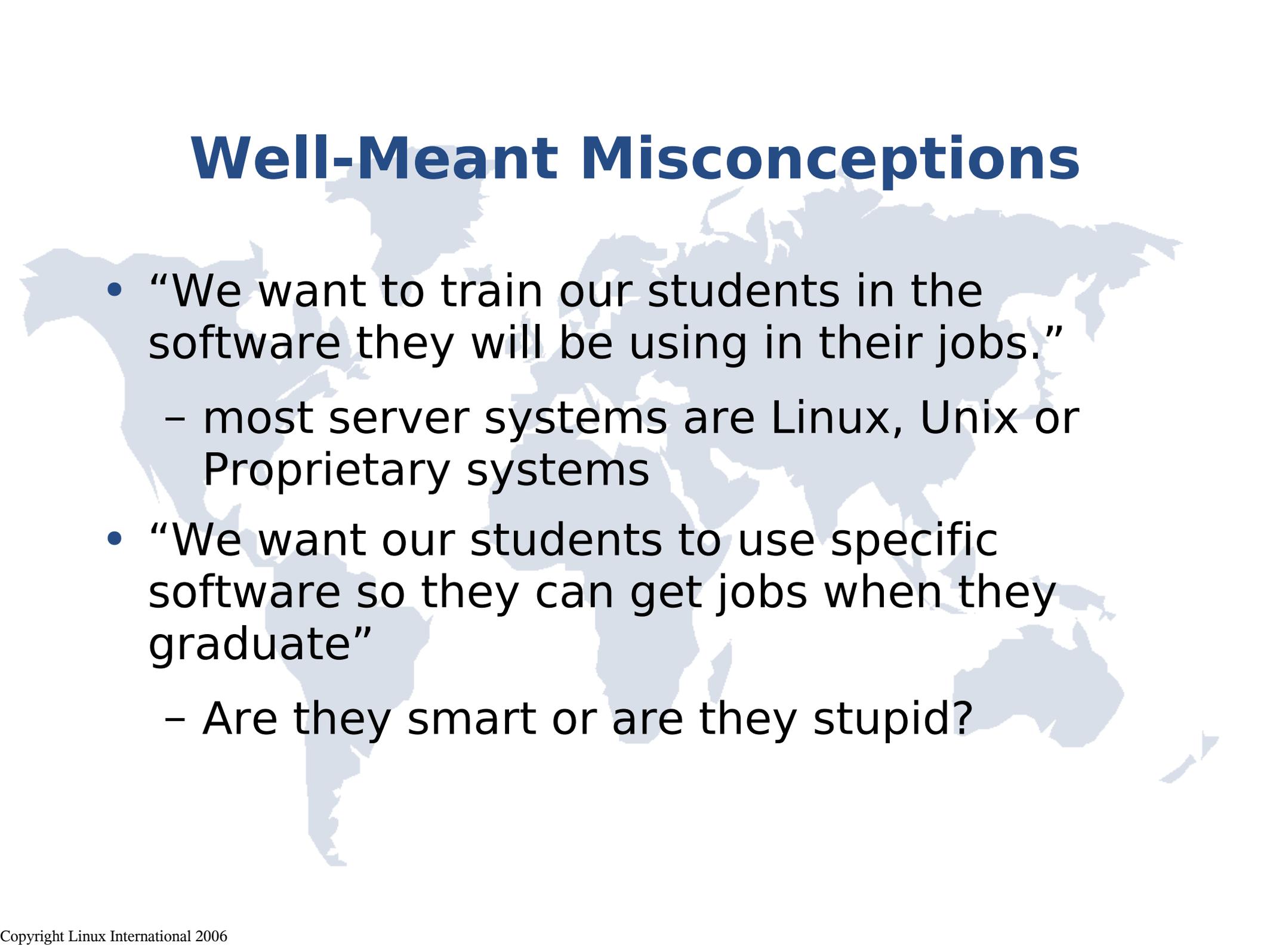
Kuali Project



Open Source financial system for Universities

- accounts
- general ledger
- purchasing
- cash disbursement
- etc.

Well-Meant Misconceptions

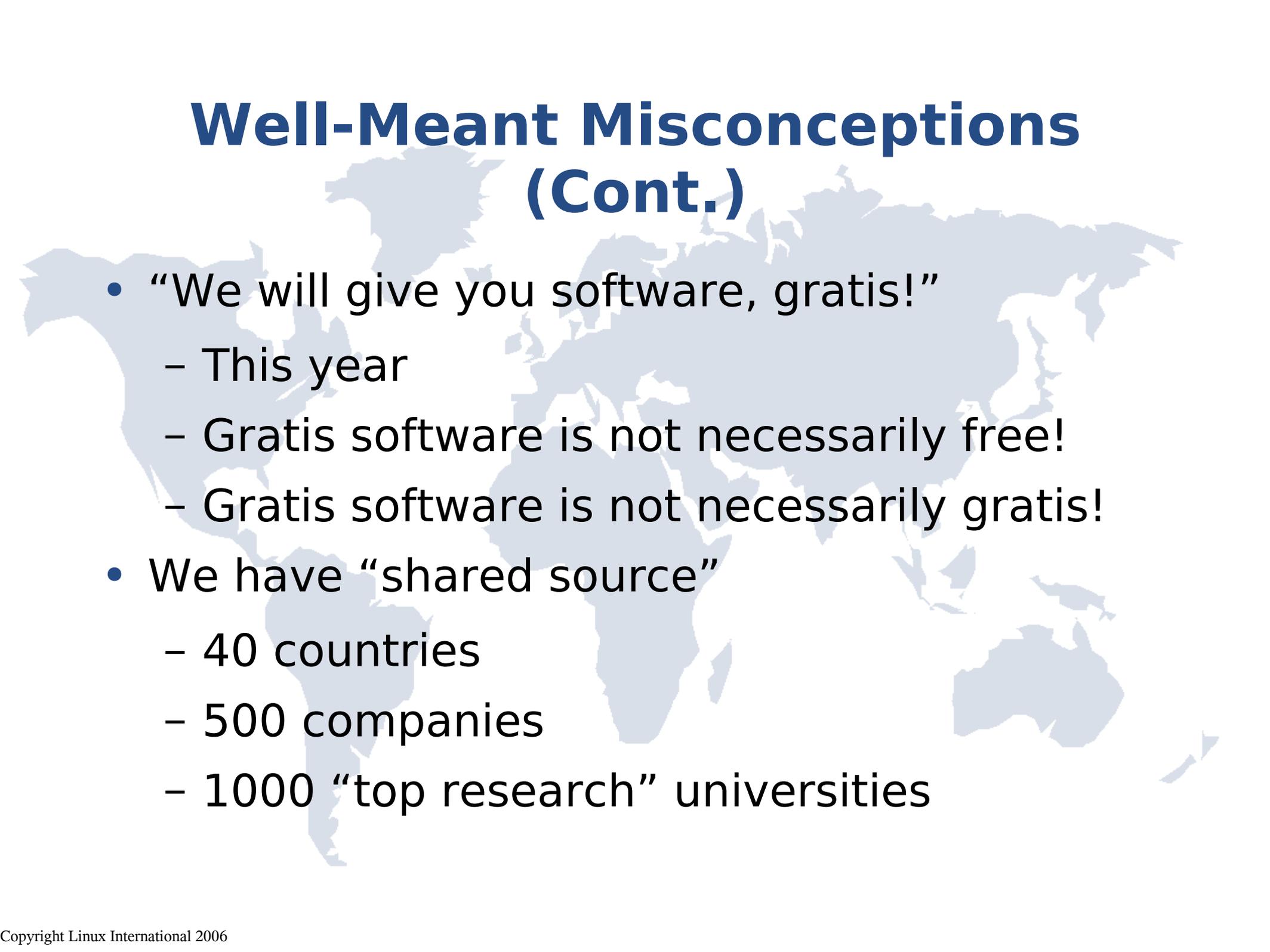


- “We want to train our students in the software they will be using in their jobs.”
 - most server systems are Linux, Unix or Proprietary systems
- “We want our students to use specific software so they can get jobs when they graduate”
 - Are they smart or are they stupid?

Well-Meant Misconceptions (Cont.)

- “We want to use widely-used software in our research so our research can be used right away by millions of people.”
 - 64 bit and clusters
- “Everybody uses XXXXXXXX”
 - Collaboration is easier
 - Except you can not change software
 - You can not collaborate with everyone

Well-Meant Misconceptions (Cont.)



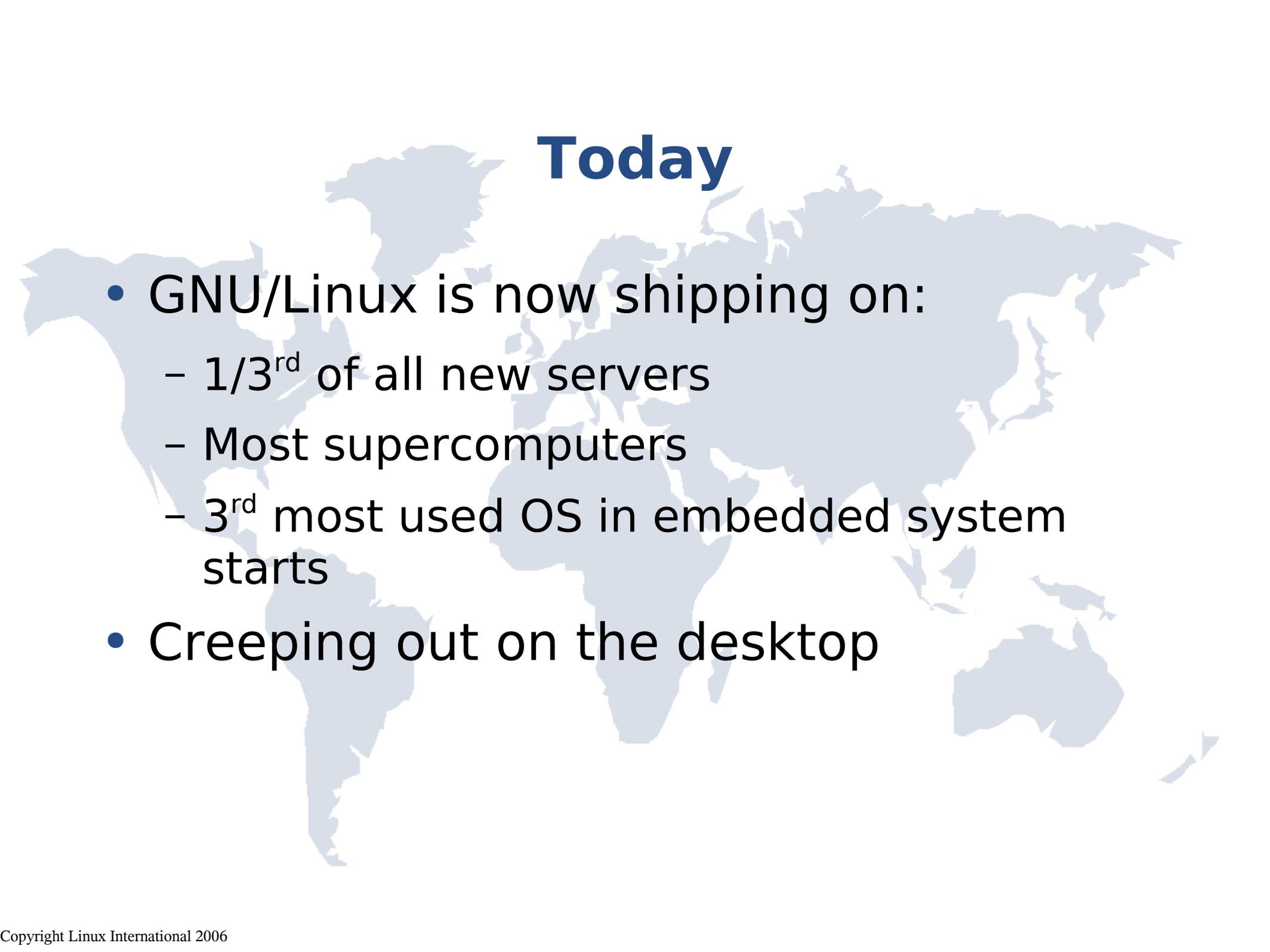
- “We will give you software, gratis!”
 - This year
 - Gratis software is not necessarily free!
 - Gratis software is not necessarily gratis!
- We have “shared source”
 - 40 countries
 - 500 companies
 - 1000 “top research” universities

Well-Meant Misconceptions (Cont.)

- “We will let you look for Trojan Horses”
 - Just like Saddam.....
- “Total Cost of Ownership Is The Same”
 - True, to a point
 - Total Cost of Ownership is not important
 - Value of the software is the real figure

Free Software is priceless!

Today



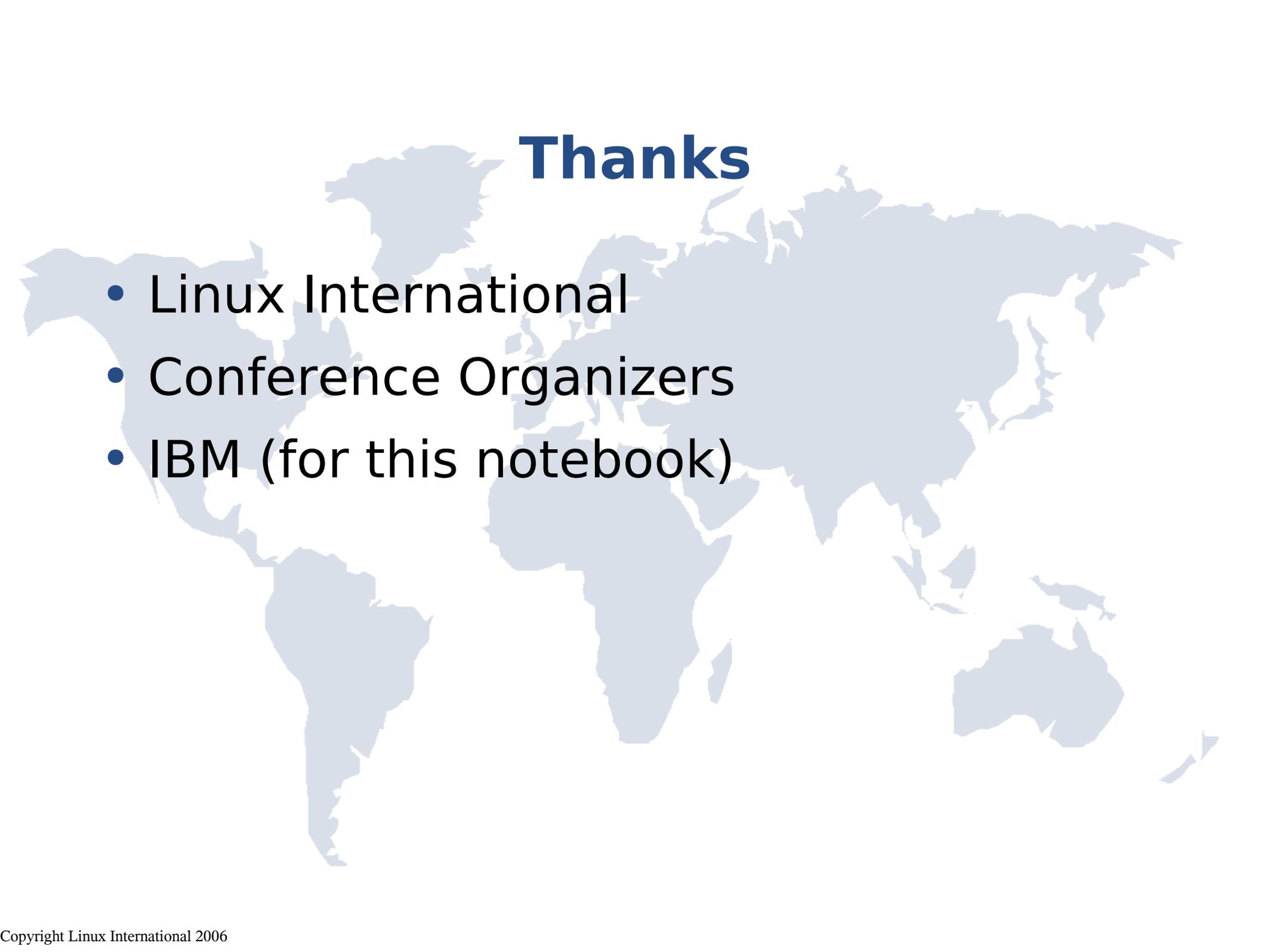
- GNU/Linux is now shipping on:
 - 1/3rd of all new servers
 - Most supercomputers
 - 3rd most used OS in embedded system starts
- Creeping out on the desktop

Summary

Free and Open Source Software Allows

- Students to see how software works
- Easy exchange of research results
- Creation of a local software economy

Thanks



- Linux International
- Conference Organizers
- IBM (for this notebook)

Questions?