

# Debian - The Universal Operating System?

Do we provide what users need?  
Do users get what they really want?

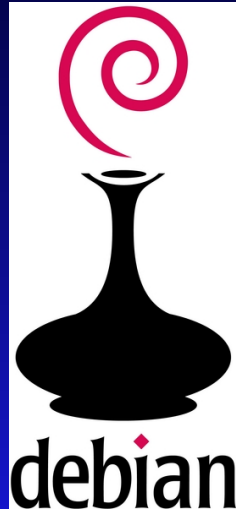
Andreas Tille

Debian Conference 7

Edinburgh, 22. June, 2007

# Overview

- 1 Different views onto Debian
- 2 Universal ?
- 3 Structure of Debian development
- 4 Deriving versus staying Debian internal
- 5 Supporting very special applications
- 6 Future



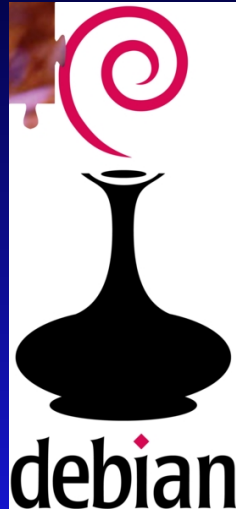
# Poor users view

- Why does the friend of my son does not install Debian?

→ Because his father is no DD.

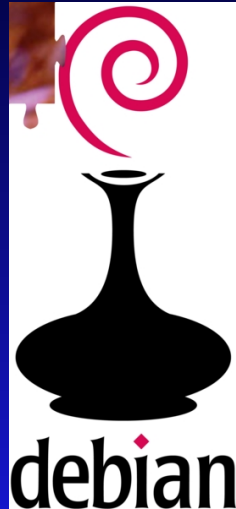
- Why do multimedia enthusiasts not prefer Debian?

→ We have no real chance to help them.



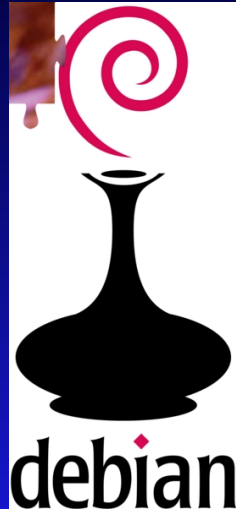
# Poor users view

- Why does the friend of my son does not install Debian?
- Because his father is no DD.
- Why do multimedia enthusiasts not prefer Debian?
- We have no real chance to help them.



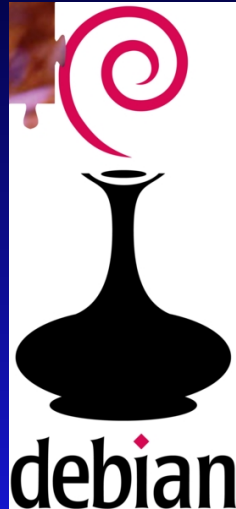
# Poor users view

- Why does the friend of my son does not install Debian?
- Because his father is no DD.
- Why do multimedia enthusiasts not prefer Debian?
- We have no real chance to help them.



# Poor users view

- Why does the friend of my son does not install Debian?
- Because his father is no DD.
- Why do multimedia enthusiasts not prefer Debian?
- We have no real chance to help them.



# Lucky users view

- Lucky to have obtained supported hardware
- Finally reached tasksel
- And now?



# Lucky users view

- Lucky to have obtained supported hardware
- Finally reached tasksel
- And now?





# Lucky users view

- Lucky to have obtained supported hardware
- Finally reached tasksel
- And now?



# Upstream developer view

- Vehicle to bring my software to the user
- Quality assurance instance
- WTFM
- Ports to other architectures



# Upstream developer view

- Vehicle to bring my software to the user
- Quality assurance instance
- WTFM
- Ports to other architectures



# Upstream developer view

- Vehicle to bring my software to the user
- Quality assurance instance
- WTFM
- Ports to other architectures



# Upstream developer view

- Vehicle to bring my software to the user
- Quality assurance instance
- WTFM
- Ports to other architectures



# Developer view

- Missing link between upstream and user
- Principles of Free Software at distribution level:  
Independence of commercial interest of producer
- Hunter and collector of Free Software  
... or rather ...
- Designer of comprehensive system



# Developer view

- Missing link between upstream and user
- Principles of Free Software at distribution level:  
Independence of commercial interest of producer
- Hunter and collector of Free Software  
... or rather ...
- Designer of comprehensive system



# Developer view

- Missing link between upstream and user
- Principles of Free Software at distribution level:  
Independence of commercial interest of producer
- Hunter and collector of Free Software  
... or rather ...
- Designer of comprehensive system





# Developer view

- Missing link between upstream and user
- Principles of Free Software at distribution level:  
Independence of commercial interest of producer
- Hunter and collector of Free Software  
...or rather ...
- Designer of comprehensive system



# Derivers view

- Debian is nice, but fits not really my needs
- Nice starting point to do something else



# Derivers view

- Debian is nice, but fits not really my needs
- Nice starting point to do something else



# Universal

WordNet: Similarity of adj universal

- 1 comprehensive (vs. noncomprehensive)
- 2 general (vs. specific)
- 3 **adaptable** (vs. unadaptable)



# Universal (2)

WordNet: The adj universal has 3 senses

- 1 cosmopolitan, ecumenical, oecumenical, general, universal, worldwide, world-wide
- 2 applicable to or common to all members of a group or set
- 3 adapted to various purposes, sizes, forms, operations



# Example 1: Teachers point of view

- handle student accounts
  - enable students to run educational software
  - teach students basic skills in information science
  - ...
- adapted from teachers point of view, but ...



# Example 1: Teachers point of view

- handle student accounts
  - enable students to run educational software
  - teach students basic skills in information science
  - ...
- adapted from teachers point of view, but ...



# Example 1: Teachers point of view

- handle student accounts
- enable students to run educational software
- teach students basic skills in information science

• ...

→ adapted from teachers point of view,  
but ...





# Example 1: Teachers point of view

- handle student accounts
- enable students to run educational software
- teach students basic skills in information science
- ...

→ adapted from teachers point of view,  
but ...



# Example 1: Teachers point of view

- handle student accounts
  - enable students to run educational software
  - teach students basic skills in information science
  - ...
- adapted from teachers point of view, but ...



## Example 2: Scientists point of view

- very interesting applications packaged
- potentially scientific workbench
- served in pieces
- no concept to guide scientific users
- only *adaptable* from scientists point of view
- slightly better for biologists



## Example 2: Scientists point of view

- very interesting applications packaged
- potentially scientific workbench
- served in pieces
- no concept to guide scientific users
- only *adaptable* from scientists point of view
- slightly better for biologists



## Example 2: Scientists point of view

- very interesting applications packaged
  - potentially scientific workbench
  - served in pieces
  - no concept to guide scientific users
- only *adaptable* from scientists point of view
- slightly better for biologists



## Example 2: Scientists point of view

- very interesting applications packaged
  - potentially scientific workbench
  - served in pieces
  - no concept to guide scientific users
- only *adaptable* from scientists point of view
- slightly better for biologists



## Example 2: Scientists point of view

- very interesting applications packaged
  - potentially scientific workbench
  - served in pieces
  - no concept to guide scientific users
- only *adaptable* from scientists point of view
- slightly better for biologists



## Example 2: Scientists point of view

- very interesting applications packaged
  - potentially scientific workbench
  - served in pieces
  - no concept to guide scientific users
- only *adaptable* from scientists point of view
- slightly better for biologists





# Do we scale well?

- 1 Number of involved people
- 2 Number of packages
- 3 Number of architectures
- 4 Number of bugs
- 5 Number of users
- 6 Number of derivatives
- 7 Number of flame wars



# Do we scale well?

- 1 Number of involved people
- 2 Number of packages
- 3 Number of architectures
- 4 Number of bugs
- 5 Number of users
- 6 Number of derivatives
- 7 Number of flame wars



# Do we scale well?

- 1 Number of involved people
- 2 Number of packages
- 3 Number of architectures
- 4 Number of bugs
- 5 Number of users
- 6 Number of derivatives
- 7 Number of flame wars



# Do we scale well?

- 1 Number of involved people
- 2 Number of packages
- 3 Number of architectures
- 4 Number of bugs
- 5 Number of users
- 6 Number of derivatives
- 7 Number of flame wars



# Do we scale well?

- 1 Number of involved people
- 2 Number of packages
- 3 Number of architectures
- 4 Number of bugs
- 5 Number of users
- 6 Number of derivatives
- 7 Number of flame wars



# Do we scale well?

- 1 Number of involved people
- 2 Number of packages
- 3 Number of architectures
- 4 Number of bugs
- 5 Number of users
- 6 Number of derivatives
- 7 Number of flame wars



# Do we scale well?

- 1 Number of involved people
- 2 Number of packages
- 3 Number of architectures
- 4 Number of bugs
- 5 Number of users
- 6 Number of derivatives
- 7 Number of flame wars



# Sub-structuring Debian

- Change of quantity into quality (Hegel)
  - To determine at the right moment the critical point where quantity changes into quality is one of the most important and difficult tasks in all the spheres of knowledge (Trotzki)
  - Evolution . . . through quantitative transformations passing into qualitative (Darwin principle)
- CDDs can deal with nearly all dimensions of growth inside Debian





# Sub-structuring Debian

- Change of quantity into quality (Hegel)
  - To determine at the right moment the critical point where quantity changes into quality is one of the most important and difficult tasks in all the spheres of knowledge (Trotzki)
  - Evolution . . . through quantitative transformations passing into qualitative (Darwin principle)
- CDDs can deal with nearly all dimensions of growth inside Debian



# Sub-structuring Debian

- Change of quantity into quality (Hegel)
- To determine at the right moment the critical point where quantity changes into quality is one of the most important and difficult tasks in all the spheres of knowledge (Trotzki)
- Evolution . . . through quantitative transformations passing into qualitative (Darwin principle)

→ CDDs can deal with nearly all dimensions of growth inside Debian



# Sub-structuring Debian

- Change of quantity into quality (Hegel)
  - To determine at the right moment the critical point where quantity changes into quality is one of the most important and difficult tasks in all the spheres of knowledge (Trotzki)
  - Evolution . . . through quantitative transformations passing into qualitative (Darwin principle)
- CDDs can deal with nearly all dimensions of growth inside Debian



# Debian is a Bazaar of Cathedrals

*For the most part, Debian is a Bazaar of Cathedrals; with a few procedures in place to override the low level cathedral in exceptional situations. Each developer has, within reason, full control over his packages, modulo following Debian Technical policy, thus creating the low level cathedral. The Technical committee, and the General Resolution Protocol offer means of overriding developer decisions about their own packages.*

-- Manoj Srivastava; 8 Oct 2003



# Structure of Debian development

- ① Originally: 1:1 (maintainer:package)
- ② Group maintenance: n:1 (maintainer:package)
  - complex packaging (X strike force)
  - reliable team work
  - stays on technical level
- ③ CDD: n:m (maintainer:package)
  - complete suites covering use cases



# Structure of Debian development

- ① Originally: 1:1 (maintainer:package)
- ② Group maintenance: n:1 (maintainer:package)
  - complex packaging (X strike force)
  - reliable team work
  - stays on technical level
- ③ CDD: n:m (maintainer:package)
  - complete suites covering use cases



# Structure of Debian development

- ① Originally: 1:1 (maintainer:package)
- ② Group maintenance: n:1 (maintainer:package)
  - complex packaging (X strike force)
  - reliable team work
  - stays on technical level
- ③ CDD: n:m (maintainer:package)
  - complete suites covering use cases



# Structure of Debian development

- ① Originally: 1:1 (maintainer:package)
- ② Group maintenance: n:1 (maintainer:package)
  - complex packaging (X strike force)
  - reliable team work
  - stays on technical level
- ③ CDD: n:m (maintainer:package)
  - complete suites covering use cases





# Structure of Debian development

- ① Originally: 1:1 (maintainer:package)
- ② Group maintenance: n:1 (maintainer:package)
  - complex packaging (X strike force)
  - reliable team work
  - stays on technical level
- ③ CDD: n:m (maintainer:package)
  - complete suites covering use cases



# Structure of Debian development

- 1 Originally: 1:1 (maintainer:package)
- 2 Group maintenance: n:1 (maintainer:package)
  - complex packaging (X strike force)
  - reliable team work
  - stays on technical level
- 3 CDD: n:m (maintainer:package)
  - complete suites covering use cases



# Structure of Debian development

- 1 Originally: 1:1 (maintainer:package)
- 2 Group maintenance: n:1 (maintainer:package)
  - complex packaging (X strike force)
  - reliable team work
  - stays on technical level
- 3 CDD: n:m (maintainer:package)
  - complete suites covering use cases



# CDD scales well

Problem

CDD-solution



# CDD scales well

Problem

CDD-solution

No. people	Smaller projects	Focussing work better control
------------	------------------	----------------------------------



# CDD scales well

Problem		CDD-solution
No. people	Smaller projects	Focussing work better control
No. packages	Subset	Closed package set



# CDD scales well

Problem		CDD-solution
No. people	Smaller projects	Focussing work better control
No. packages	Subset	Closed package set
No. bugs	CDD relevant	Stronger focus



# CDD scales well

Problem		CDD-solution
No. people	Smaller projects	Focussing work better control
No. packages	Subset	Closed package set
No. bugs	CDD relevant	Stronger focus
No. users	Specialists	Interested users known needs





# CDD scales well

Problem		CDD-solution
No. people	Smaller projects	Focussing work better control
No. packages	Subset	Closed package set
No. bugs	CDD relevant	Stronger focus
No. users	Specialists	Interested users known needs
No. derivatives	Customisation	Keep inside



# CDD scales well

Problem		CDD-solution
No. people	Smaller projects	Focussing work better control
No. packages	Subset	Closed package set
No. bugs	CDD relevant	Stronger focus
No. users	Specialists	Interested users known needs
No. derivatives	Customisation	Keep inside
Time span	Releases	Independent



# Number of Derivatives

Debian	Fedora	Others
129	63	<30



# Number of Derivatives

Debian	Fedora	Others
129	63	<30



# The poor, diligent deriver

- Misconception: Debian is one-way
- Conflicts with human beings (yes we are ☺)
- Keep derivative continuously up to date
- Rather try to cooperate



# The poor, diligent deriver

- Misconception: Debian is one-way
- Conflicts with human beings (yes we are ☺)
- Keep derivative continuously up to date
- Rather try to cooperate



# The poor, diligent deriver

- Misconception: Debian is one-way
- Conflicts with human beings (yes we are ☺)
- Keep derivative continuously up to date
- Rather try to cooperate



# The poor, diligent deriver

- Misconception: Debian is one-way
- Conflicts with human beings (yes we are ☺)
- Keep derivative continuously up to date
- Rather try to cooperate





# The impatient, desperate deriver

- Show stoppers in Debian
- Deriver has timeline
- DoOcracy
- Hint: Leave open the way back



# The impatient, desperate deriver

- Show stoppers in Debian
- Deriver has timeline
- DoOcracy
- Hint: Leave open the way back



# The impatient, desperate deriver

- Show stoppers in Debian
- Deriver has timeline
- DoOcracy
- Hint: Leave open the way back



# The impatient, desperate deriver

- Show stoppers in Debian
- Deriver has timeline
- DoOcracy
- Hint: Leave open the way back



# The authority employed deriver

- LiMux, Wienux, LinEx, Lliurex, ...
- Authorities are something else
- Alternatively Debian-eGov
- Similar situation: Funded projects (DeMuDi)



# The authority employed deriver

- LiMux, Wienux, LinEx, Lliurex, ...
- Authorities are something else
- Alternatively *Debian-eGov*
- Similar situation: Funded projects (*DeMuDi*)



# The authority employed deriver

- LiMux, Wienux, LinEx, Lliurex, ...
- Authorities are something else
- Alternatively Debian-eGov
- Similar situation: Funded projects (DeMuDi)



# The authority employed deriver

- LiMux, Wienux, LinEx, Lliurex, ...
- Authorities are something else
- Alternatively Debian-eGov
- Similar situation: Funded projects (DeMuDi)





# The lucky live CD creator

- Live CD cool
  - Demonstrating / showing off your nice product
  - Testing your hardware
  - Boot your favourite OS on random computer
  - Kiosk system
  - Linux course in a random computer lab
- Problem: Keep read-only system up to date
- Use *live-helper*
- Join the effort (DoOcracy)



# The lucky live CD creator

- Live CD cool
  - Demonstrating / showing off your nice product
  - Testing your hardware
  - Boot your favourite OS on random computer
  - Kiosk system
  - Linux course in a random computer lab
- Problem: Keep read-only system up to date
  - Use *live-helper*
  - Join the effort (DoOcracy)



# The lucky live CD creator

- Live CD cool
  - Demonstrating / showing off your nice product
  - Testing your hardware
  - Boot your favourite OS on random computer
  - Kiosk system
  - Linux course in a random computer lab
- Problem: Keep read-only system up to date
- Use *live-helper*
- Join the effort (DoOcracy)



# The lucky live CD creator

- Live CD cool
  - Demonstrating / showing off your nice product
  - Testing your hardware
  - Boot your favourite OS on random computer
  - Kiosk system
  - Linux course in a random computer lab
- Problem: Keep read-only system up to date
- Use *live-helper*
- Join the effort (DoOcracy)



# The clever, commercial deriver

- Turn Debian into sellable product
- Linspire, Mephis, Progeny, Ubuntu, Xandros, ...
- Add some value (non-free drivers helping poor users, etc.)
- Sell some service (24/7 for enterprises)
- Basically re-branding
- Debian supports those derivers
- Debian competes with those derivers



# The clever, commercial deriver

- Turn Debian into sellable product
- Linspire, Mephis, Progeny, Ubuntu, Xandros, ...
- Add some value (non-free drivers helping poor users, etc.)
- Sell some service (24/7 for enterprises)
- Basically re-branding
- Debian supports those derivers
- Debian competes with those derivers



# The clever, commercial deriver

- Turn Debian into sellable product
- Linspire, Mephis, Progeny, Ubuntu, Xandros, ...
- Add some value (non-free drivers helping poor users, etc.)
- Sell some service (24/7 for enterprises)
- Basically re-branding
- Debian supports those derivers
- Debian competes with those derivers



# The clever, commercial deriver

- Turn Debian into sellable product
- Linspire, Mephis, Progeny, Ubuntu, Xandros, ...
- Add some value (non-free drivers helping poor users, etc.)
- Sell some service (24/7 for enterprises)
- Basically re-branding
- Debian supports those derivers
- Debian competes with those derivers





# The clever, commercial deriver

- Turn Debian into sellable product
- Linspire, Mephis, Progeny, Ubuntu, Xandros, ...
- Add some value (non-free drivers helping poor users, etc.)
- Sell some service (24/7 for enterprises)
- Basically re-branding
- Debian supports those derivers
- Debian competes with those derivers



# The clever, commercial deriver

- Turn Debian into sellable product
- Linspire, Mephis, Progeny, Ubuntu, Xandros, ...
- Add some value (non-free drivers helping poor users, etc.)
- Sell some service (24/7 for enterprises)
- Basically re-branding
- Debian supports those derivers
- Debian competes with those derivers



# The clever, commercial deriver

- Turn Debian into sellable product
- Linspire, Mephis, Progeny, Ubuntu, Xandros, ...
- Add some value (non-free drivers helping poor users, etc.)
- Sell some service (24/7 for enterprises)
- Basically re-branding
- Debian supports those derivers
- Debian competes with those derivers



# Derivers lessons to learn

- $\Delta(\textit{Debian} - \textit{Derivative}) < \varepsilon \ \forall t$   
In other words: Keep the diff always small!
- We are on your side.  
Talk to us.



# Derivers lessons to learn

- $\Delta(\textit{Debian} - \textit{Derivative}) < \varepsilon \ \forall t$   
In other words: Keep the diff always small!
- We are on your side.  
Talk to us.



# Pros and cons for special applications

- Debian becomes larger and larger
  - How many packages are good for Debian?
    - Bring special applications under quality control
    - Attract wider user base
  - How many packages are good for our users?
    - Large user data (why `#38902 wont-fix`)



# Pros and cons for special applications

- Debian becomes larger and larger
- How many packages are good for Debian?
  - Bring special applications under quality control
  - Attract wider user base
- How many packages are good for our users?
  - Large user data (why `#38902 wont-fix`)



# Pros and cons for special applications

- Debian becomes larger and larger
- How many packages are good for Debian?
  - Bring special applications under quality control
  - Attract wider user base
- How many packages are good for our users?
  - Large user data (why `#38902 wont-fix`)





# Pros and cons for special applications

- Debian becomes larger and larger
- How many packages are good for Debian?
  - Bring special applications under quality control
  - Attract wider user base
- How many packages are good for our users?
  - Large user data (why `#38902 wont-fix`)



# Pros and cons for special applications

- Debian becomes larger and larger
- How many packages are good for Debian?
  - Bring special applications under quality control
  - Attract wider user base
- How many packages are good for our users?
  - Large user data (why `#38902 wont-fix`)



# Pros and cons for special applications

- Debian becomes larger and larger
- How many packages are good for Debian?
  - Bring special applications under quality control
  - Attract wider user base
- How many packages are good for our users?
  - Large user data (why `#38902 wont-fix`)



# Basic goal of CDDs

- Debian > 15000 packages
- Users interested in *subset*
- Groups of specialised users
- Easy installation and configuration
- While Debian stays general support specialists as well
- **No derivative** from Debian

*Basic idea: Do not make a separate distribution but make Debian fit for special purpose instead*



# Basic goal of CDDs

- Debian > 15000 packages
- Users interested in *subset*
- Groups of specialised users
- Easy installation and configuration
- While Debian stays general support specialists as well
- **No derivative** from Debian

*Basic idea: Do not make a separate distribution but make Debian fit for special purpose instead*



# Basic goal of CDDs

- Debian > 15000 packages
- Users interested in *subset*
- Groups of specialised users
- Easy installation and configuration
- While Debian stays general support specialists as well
- **No derivative** from Debian

*Basic idea: Do not make a separate distribution but make Debian fit for special purpose instead*



# Basic goal of CDDs

- Debian > 15000 packages
- Users interested in *subset*
- Groups of specialised users
- Easy installation and configuration
- While Debian stays general support specialists as well
- **No derivative** from Debian

*Basic idea: Do not make a separate distribution but make Debian fit for special purpose instead*



# Basic goal of CDDs

- Debian > 15000 packages
- Users interested in *subset*
- Groups of specialised users
- Easy installation and configuration
- While Debian stays general support specialists as well
- **No derivative** from Debian

*Basic idea: Do not make a separate distribution but make Debian fit for special purpose instead*





# Basic goal of CDDs

- Debian > 15000 packages
- Users interested in *subset*
- Groups of specialised users
- Easy installation and configuration
- While Debian stays general support specialists as well
- **No derivative** from Debian

*Basic idea: Do not make a separate distribution but make Debian fit for special purpose instead*



# Basic goal of CDDs

- Debian > 15000 packages
- Users interested in *subset*
- Groups of specialised users
- Easy installation and configuration
- While Debian stays general support specialists as well
- **No derivative** from Debian

*Basic idea: Do not make a separate distribution but make Debian fit for special purpose instead*



# Future of Debian

- CDD solve structural problems inside Debian
- Fit user interests
- Makes Debian stronger if done the right way

*Also known as: The last, final step  
towards Total World Domination!*



# Future of Debian

- CDD solve structural problems inside Debian
- Fit user interests
- Makes Debian stronger if done the right way

*Also known as: The last, final step  
towards Total World Domination!*



# Future of Debian

- CDD solve structural problems inside Debian
- Fit user interests
- Makes Debian stronger if done the right way

*Also known as: The last, final step  
towards Total World Domination!*



# Future of Debian

- CDD solve structural problems inside Debian
- Fit user interests
- Makes Debian stronger if done the right way

*Also known as: The last, final step  
towards Total World Domination!*



This talk can be found at

<http://people.debian.org/~tille/talks/>  
Andreas Tille <tille@debian.org>

