

Free software: How does it work?

European Broadcasting Union Seminar

Geneva,

October 1, 2007

Rishab Aiyer Ghosh (ghosh@merit.unu.edu)

United Nations University / MERIT, Maastricht

Software: the object

10101010101111100101001110101010010101011
10001110101010010101011101010101010111110
01000111010101001010101110101010101011111
00101001110101010010101011101010101010111
11001010011101010100101010111010101010101
11110010100111010101001010101110101010101
01111100101001110101010010101011101010101
01011111001010011101010100101010111010101
01010111110010100111010101001010101110101
01010101111100101001110101010010101011101
01010101011111001010011101010100101010111
01010101010111110010100111010101001010101
11010101010101111100101001110101010010111

Software: source code

```
// "hello world" program
// use standard input-output functions
#include <stdio.h>

void main () {
    printf("hello, world!\n"); // print it!
}
```

Software: functionality

```
hello world!
```

Free/Libre/Open Source Software

Software which users have freedoms to:

- Use (the software's functionality)

- Study (the software source code)

- Distribute (at any or no price)

- Change (and distribute changes)

Free as in Freedom

Free software

=

Libre software

=

Open source software

(different terms for the *same software*)

Free as in Freedom

Proprietary software

=

closed source

software *without* the freedom to use,
study, share and modify

Source vs Standards

Open standards: easily implemented by any software, without legal, technical, financial barriers

Source vs Standards

Open standards: easily implemented by any software, without legal, technical, financial barriers

Other standards (not open!): proprietary standards, or standards which different producers may be prevented from implementing for legal, technical or financial reasons

Source vs Standards

Open source uses open standards

Source vs Standards

Open source uses open standards

Proprietary software often uses closed standards

Source vs Standards

Open source uses open standards

Proprietary software often uses closed standards

Open standards limit vendor lock-in

Source vs Standards

Open source uses open standards

Proprietary software often uses closed standards

Open standards limit vendor lock-in

HTML – any software can read the web

SMTP – any software can send email

Source vs Standards

Open source uses open standards

Proprietary software often uses closed standards

Open standards limit vendor lock-in

HTML – any software can read the web

SMTP – any software can send email

DOC – works best/only with Microsoft

Source vs Standards

Closed standards and broadcasting:

Stream only in Windows Media

=

force your audience to pay Microsoft



+ users have control

+

+ users have control

+ adapt software to your needs



- + users have control
 - + adapt software to your needs
 - + no need to change hardware

+

- + users have control
 - + adapt software to your needs
 - + no need to change hardware
 - + decide when and how to change



- + users have control
 - + adapt software to your needs
 - + no need to change hardware
 - + decide when and how to change
 - + no forced software upgrades



- + users have control
 - + adapt software to your needs
 - + no need to change hardware
 - + decide when and how to change
 - + no forced software upgrades
 - + no vendor lock-in

+

+ users have control

+ reduce costs

+

+ users have control

+ reduce costs

+ in-house support (if skills exist)

+

- + users have control
- + reduce costs
 - + in-house support (if skills exist)
 - + external support of *your* choice

+

+ users have control

+ reduce costs

+ in-house support (if skills exist)

+ external support of *your* choice

+ adaptable to cheaper hardware

+

+ users have control

+ reduce costs

+ in-house support (if skills exist)

+ external support of *your* choice

+ adaptable to cheaper hardware

+ can reduce running costs

+

- + users have control
- + reduce costs
- + open standards

+

- + users have control
- + reduce costs
- + open standards
 - + no proprietary standards

+

- + users have control
- + reduce costs
- + open standards
 - + no proprietary standards
 - + give your customers choice

+

- + users have control
- + reduce costs
- + open standards
 - + no proprietary standards
 - + give your customers choice
 - + increased interoperability



- + users have control
- + reduce costs
- + open standards
- + sustainability

+

- + users have control
- + reduce costs
- + open standards
- + sustainability
 - + code forever accessible



- + users have control
- + reduce costs
- + open standards
- + sustainability
- + skills development

+

- + users have control
- + reduce costs
- + open standards
- + sustainability
- + skills development
- + #1 reason for developers



- + users have control
- + reduce costs
- + open standards
- + sustainability
- + skills development
 - + #1 reason for developers
 - + FLOSS is a training environment



- + users have control
- + reduce costs
- + open standards
- + sustainability
- + skills development
- + security and quality

– / ?

– smaller installed base

– / ?

- smaller installed base
 - fewer users = less support

– / ?

- smaller installed base
 - fewer users = less support
 - ? but support increasing with users

– / ?

- smaller installed base
 - fewer users = less support
 - ? but support increasing with users
 - + niche areas have a lot of support

– / ?

- smaller installed base
- compatibility with proprietary apps

– / ?

- smaller installed base
- compatibility with proprietary apps
 - ? prop vendors want to lock you in

– / ?

- smaller installed base
- compatibility with proprietary apps
- who can i blame?

– / ?

- smaller installed base
- compatibility with proprietary apps
- who can i blame?

? not your *proprietary* vendor!

? service contracts for liability

– / ?

- smaller installed base
 - compatibility with proprietary apps
 - who can i blame?
 - legal issues?
- ? have you read your prop. licence?

– / ?

- smaller installed base
- compatibility with proprietary apps
- who can i blame?
- legal issues?
- ease of use

– / ?

- smaller installed base
- compatibility with proprietary apps
- who can i blame?
- legal issues?
- ease of use
 - mainly for end-user apps
- + rapidly improving

– / ?

- smaller installed base
- compatibility with proprietary apps
- who can i blame?
- legal issues?
- ease of use

Outlook

- < 20% of software is bought as a product
- Rest of software – custom, in-house – has an economics like open source
- “FLOSS-related” services could account for >30% of IT market by 2010

Thank you

Rishab Aiyer Ghosh

ghosh@merit.unu.edu

UNU-MERIT / FLOSS Project

European Commission report:

www.flossimpact.eu