

---

# Perché usare il modello del software libero in un contesto di ricerca

Alessandro Rubini

`<rubini@{gnu.org|linux.it|fsfeurope.org}>`

`http://www.fsfeurope.org`

`http://ar.linux.it`

- **La specificità del software come prodotto**
- **Perché usare GNU/Linux come utente**
- **Perché scrivere software usando licenze libere**
- **Il mercato del lavoro con il software libero**
- **Il problema dell'assistenza**

# Specificità dell' «oggetto» software

---

**Oggi sempre più cose si fanno con l'elaboratore**

- Questo vuol dire hardware ma soprattutto software

**L'informazione e il codice controllano la nostra vita**

- Come dice Lessig: «code is law»

**L'informazione non è un bene fisico, e il codice:**

- È informazione pura
- È un prodotto culturale
- È un prodotto scientifico
- Si copia a costo zero
- Può essere adattato alle esigenze

## **Il software è un servizio**

**Il mercato oggi gira sui servizi, ed è giusto così**

- Il mercato del bene fisico non è sostenibile

**Il mercato informatico è fortemente monopolistico**

- Questo non è il problema, solo un effetto collaterale

# Cos'è il software libero

---

**Il software libero è un modello per massimizzare la produzione e la diffusione del patrimonio intellettuale**

**0: libertà di usare, per qualsiasi scopo**

**1: libertà di studiare ed adattare**

**2: libertà di copiare**

**3: libertà di migliorare e redistribuire**

<http://www.gnu.org/philosophy/free-sw.html>

<http://www.gnu.org/philosophy/free-sw.it.html>

<http://www.gnu.org/philosophy/freedom-or-power.html>

<http://www.gnu.org/philosophy/freedom-or-power.it.html>

# Perché usare GNU/Linux come utente

---

## Costo del prodotto

Prezzo calato dall'alto  
Aggiornamenti obbligatori

Commisurato al lavoro fatto  
Aggiornamenti facoltativi

## Installazioni multiple

Non permesse, oppure onerose  
Ingrandirsi è un costo

Permesse ed incoraggiate  
Ingrandirsi non è un costo

## Esclusiva

Solo se si comprano i diritti  
In pratica spesso impossibile

Contrattabile con l'autore  
in base alle esigenze

## Disponibilità (hw, formati)

Gira solo su macchine recenti  
I dati sono prigionieri del programma

Multi piattaforma e hw obsoleto  
I formati dei dati sono accessibili

## Assistenza

Vincolata al fornitore iniziale

Ci si può rivolgere a terzi  
Si può anche non comprare

# Perché scrivere sw usando licenze libere

---

## Formazione

Corsi onerosi  
Firma di NDA

Tutta l'informazione è disponibile

## Creazione di codice

Creazione da zero  
Accordi onerosi con terzi

Si può attingere dall'esistente  
Si possono adattare altre soluzioni

## Commercializzazione

Campagna pubblicitaria  
Rete di distribuzione

Contatto personale  
Distribuzione via rete

## Personalizzazione

Necessaria per ogni cliente  
Impossibilità di delegare

Il cliente può arrangiarsi  
Si può delegare a terzi

## Assistenza

Occorre fornire assistenza

Un autore può offrire o meno  
un contratto di assistenza

# Tipi di realtà

---

Possiamo distinguere tra tre "tipi" di realtà produttive

## **Ci sono realtà che**

**utilizzano il software  
per produrre altro**

**producono software  
per venderlo**

**creano software  
per altri fini**

# Il passaggio al software libero

---

## **Per chi usa software per produrre altro**

- Indipendenza dal fornitore
- Risparmio sui costi di licenza
- Possibilità di formare personale interno
- Possibilità di farsi assistenza da solo

## **Per chi produce software per venderlo**

- Indipendenza dal cliente
- Maggiore libertà di operazione
- Miglior servizio al cliente
- Credibilità di lungo periodo
- Modelli economici basati sull'assistenza
- Modelli economici basati sulla personalizzazione
- Modelli economici basati sul «branding»

## **Per chi produce software all'interno di altre attività**

- Indipendenza dal fornitore
- Accesso diretto alla tecnologia
- Possibilità di innovare sull'esistente
- Indipendenza nelle scelte hardware

# Le critiche

---

## **Il software è sempre stato proprietario**

Falso: il software è nato libero  
La proprietarizzazione è successiva

## **Il povero programmatore deve mangiare**

Il programmatore oggi mangia già poco  
Come tutti, il programmatore vive del suo lavoro

## **Le aziende attuali sarebbero in difficoltà**

Le aziende attuali abusano del diritto d'autore  
Ogni azienda deve adattarsi al mercato

## **Dobbiamo tutelare la «proprietà intellettuale»**

La «proprietà intellettuale» non esiste  
Le nostre licenze si basano sul diritto d'autore

## **Non ci sarebbe più sviluppo**

La libera circolazione delle idee favorisce lo sviluppo  
Un mercato di libera competizione favorisce le innovazioni

## **Il mio concorrente può copiare le mie idee**

Purtroppo esistono alcuni casi in cui questo è vero;  
in particolare, chi vende un sistema «standard» con sw speciale.

<http://www.linux.it/GNU/nemici/>



# Sono discorsi utopici?

---

Esistono altri settori «a informazione disponibile» a riprova che non stiamo facendo solo demagogia?

**Il settore medico**

**Il settore legale**

**Il settore dei commercialisti**

Ciascuna con le sue peculiarità, queste sono realtà floride sia come mercato sia, dove applicabile, come ricerca.

Nonostante questi siano mercati della sola informazione, gli operatori del settore non necessitano dell'esclusiva sull'informazione che detengono e producono.

**Nessuna licenza libera impedisce la vendita del prodotto**

# Il problema dell'assistenza

---

## **Tutta l'informazione è disponibile**

Le aziende sono incentivate a formare personale interno  
Per necessità specialistiche occorrerà rivolgersi comunque sul mercato

## **Il sistema favorisce la competenza distribuita sul territorio**

È più facile trovare il professionista che la grande azienda  
Si pensi ancora al parallelo legale o medico  
**Il mercato è ancora giovane**

Il distributore non può pagare la struttura di supporto con i margini sulla vendita  
Perché fiorisca l'offerta di assistenza occorre la domanda

## **«Se il software è modificabile, l'assistenza è impossibile»**

I contratti di assistenza devono presupporre che il cliente non tocchi il codice  
Chi modifica il software ha le competenze per assistersi da sé  
**«Solo l'autore può offrire assistenza qualificata»**

Quando l'autore non la offre possono nascere altre realtà commerciali all'uopo  
Lasciamo che l'autore faccia l'autore e sia remunerato per quello

# La «tutela legale» dei programmi

---

**Chi crea prodotti di mercato vuole l'esclusiva**

Tramite il diritto d'autore

Tramite il segreto industriale

Tramite il brevetto industriale

**I tre tipi di esclusiva sono incompatibili, eppure:**

I programmi sono opere autoriali

I programmi sono assolutamente segreti

Si vogliono brevettare le idee in un programma

**Il pericolo più grosso per le imprese europee oggi  
è l'avvento della brevettabilità delle idee astratte,  
il cosiddetto «brevetto software»**

# Le licenze di diritto d'autore

Ci sono due categorie principali di licenze libere

- Licenze persistenti, o «copyleft» (permesso d'autore)
  - ♦ La redistribuzione del sw deve redistribuire anche i permessi associati
  - ♦ Ogni lavoro derivato deve mantenere i permessi associati.
- ♦ Esempi: licenza GNU GPL, GNU LGPL
- Licenze non persistenti
  - ♦ Si permette la redistribuzione sotto termini diversi
  - ♦ Esempi: licenza MIT, BSD.

**Quale forma di licenza utilizzare per il proprio lavoro?**

- **Le licenze persistenti**
  - ♦ mantengono la libertà nel tempo
  - ♦ permettono ai piccoli di non veder fagocitato il proprio lavoro
- **Le licenze non persistenti**
  - ♦ permettono agli investitori di estendere in forma proprietaria
  - ♦ permettono ai gruppi di sviluppo di entrare prima sul mercato

# Come applicare la licenza al proprio lavoro

**Il diritto d'autore nasce in capo alla creazione dell'opera**

- Per applicare una licenza basta dichiararlo nel programma
- Alcune licenze suggeriscono il testo da includere
- Tale dichiarazione non ha valore probatorio

**Cose a cui fare attenzione**

- La compatibilità tra le diverse licenze
  - ◊ Le licenze per il codice che viene riutilizzato
  - ◊ Come si vuole che sia riutilizzato il proprio lavoro
- La chiarezza nell'applicazione dei termini di licenza
  - ◊ Un'opera senza licenza non è redistribuibile
  - ◊ Diversi componenti di un pacchetto posso avere licenze diverse
- Essere realmente i titolari dei diritti
  - ◊ I diritti sul lavoro di un dipendente sono del datore di lavoro

# Cosa fare in caso di violazione dei diritti?

## **Quando si verifica la violazione dei diritti di altri autori**

- Comunicarlo all'autore del lavoro illecitamente distribuito
- Comunicarlo a FSF/FSFE

## **Quando si verifica la violazione dei propri diritti**

- Raccogliere quante più informazioni sulla violazione
- Comunicarlo a FSF/FSFE

## **E in pratica?**

- Un legale notifica la violazione all'azienda colpevole
- La violazione viene eliminata
- Si evita di pubblicizzare l'avvenuta violazione.

## **Nessuna violazione finora è stata portata in tribunale**

- Ogni violazione avvenuta finora è stata risolta pacificamente
- Le aziende riconoscono la validità delle licenze di software libero

# Conclusioni e digressioni

---

**Tutte le aziende (tranne due o tre) dovrebbero prendere in considerazione il passaggio al software libero**

**La situazione attuale del mercato sw è senza sbocchi**

**Per chi non è monopolista, seguirne il modello è una tattica perdente**

**Nuove forme di distribuzione richiedono nuovi modelli economici**

**Ovviamente, cambiare modello comporta costi di transizione**

**Non è una rivoluzione, non è demagogia, non è comunismo.**

**Occorre fare molta attenzione alla scelta delle licenze  
(«persistenti» e non, libere e semi-libere)**

**Le argomentazioni sociali non vanno dimenticate,  
anche se in questo contesto risultano fuori tema**