

---

# Introduzione al Software Libero

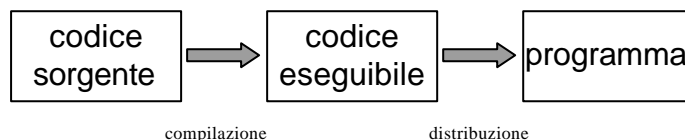
Linux Day 6  
23 novembre 2002

---

## Il software

---

- Programma (codice) scritto in un particolare linguaggio di programmazione
  - esistono molti linguaggi di programmazione
- (Di solito) è diviso in due parti
  - codice sorgente (parte sviluppata dal programmatore)
  - codice eseguibile (parte utilizzata dall'utente)
    - normalmente in forma binaria
      - (se scritto con un linguaggio compilato)



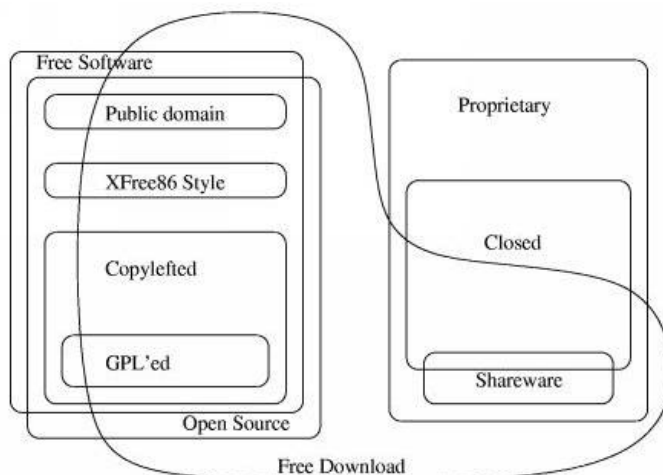
## Licenze del software (1)

---

- I programmi (di solito) sono tutelati dalle leggi sul diritto di autore
  - la licenza rappresenta il “contratto d’uso”
  - il copyright corrisponde ad avere i diritti d’ autore
- I programmi possono essere distribuiti in varie forme
  - gratuiti o a pagamento
  - con o senza sorgente
  - con o senza licenza
- Esistono diversi tipi di licenze software

## Licenze del software (2)

---



<http://www.gnu.org/philosophy/categories.it.html>

## Software Libero (*Free Software*)

---

- Libertà dell'utente di eseguire, copiare, distribuire, studiare, cambiare e migliorare il software
- Diversi tipi di software libero
  - Software **GPL**
  - Software libero con **copyleft**
    - le eventuali modifiche devono essere rilasciate come software libero
  - Software libero senza **copyleft**
    - non garantisce che le modifiche possano essere rilasciate come software non libero
  - Software di pubblico dominio
    - software libero senza copyright
    - a completa disposizione di chiunque

## Le libertà del Software Libero

---

- Libertà di eseguire il programma, per qualsiasi scopo
- Libertà di studiare come funziona il programma, e adattarlo alle proprie necessità
  - l'accesso al codice sorgente ne è un prerequisito
- Libertà di ridistribuire copie in modo da aiutare il prossimo
- Libertà di migliorare il programma, e distribuirne pubblicamente i miglioramenti, in modo tale che tutta la comunità ne tragga beneficio
  - l'accesso al codice sorgente ne è un prerequisito

## Conseguenze

---

- Queste libertà richiedono l'accessibilità del codice sorgente
  - ma non è condizione sufficiente
- Possono essere imposte restrizioni per garantire queste libertà (**copyleft**)
- Queste libertà non si applicano solo al software ma alla produzione culturale in genere
  
- Per maggiori informazioni:
  - <http://www.gnu.org/philosophy/free-sw.it.html>

## Vantaggi derivati

---

- Il prezzo (*ma non è detto*)
  - si investe non nel prodotto ma nel know-how
- Indipendenza dal fornitore
  - possibilità di ricorrere a qualunque esperto
- Verificabilità del comportamento del codice
  - importante nel caso di documenti riservati e questioni di sicurezza
- Accesso alla tecnologia
- Riutilizzo del codice
- Patrimonio pubblico
  - i miglioramenti sono a disposizione di tutti

# Software non libero

---

- Software semi-libero
  - può prevedere solo alcune delle libertà di uso, copia, distribuzione o modifica
- Software proprietario
  - l'uso, la modifica e la distribuzione sono proibite o fortemente ristrette, ma può essere gratuito
- Freeware
  - software gratuito, non comporta né l'accesso ai sorgenti né la possibilità di redistribuzione
- Shareware
  - software di cui è permessa la distribuzione
  - ma bisogna pagare per l'uso

# Software commerciale

---

- Software commerciale
  - software venduto e per il quale c'è assistenza commerciale
  - *anche il software libero può essere commerciale*
    - la discriminante NON è il costo ma il tipo di licenza!

# OpenSource o Software Libero

---

- Differenze sul piano filosofico, non su quello pratico (*almeno per ora*)
- OpenSource
  - nato per evitare l'ambiguità del termine “*free*”
  - mette l'accento sugli aspetti di convenienza pratica
  - tende a tralasciare gli aspetti filosofici (legati alla libertà)
- Software Libero
  - mette l'accento sull'aspetto filosofico
  - si tende a salvaguardare la libertà al di là degli aspetti di convenienza immediata

# Copyright e Copyleft

---

- Copyright
  - traducibile come *diritto d'autore*
    - che però ha un significato più ampio
  - indica una serie di diritti che l'autore in genere trattiene e non concede agli utenti
- Copyleft
  - tradotto come *permesso d'autore*
  - è una forma speciale di uso del copyright in cui si cedono i diritti agli utenti, impedendo che altri possano ritirarli in seguito
  - <http://www.gnu.org/copyleft/copyleft.html>

# Licenze per il Software Libero (1)

---

- Software libero con copyleft
  - GNU GPL (General Public Licence)
    - usa il diritto d'autore per proteggere la libertà del software
    - impone che i lavori che usano codice GPL vengano rilasciati con la stessa licenza
    - protegge la libertà del software, tutelando:
      - gli autori (a cui resta il copyright)
      - gli utenti (a cui restano le libertà fondamentali)
  - GNU LGPL (Lesser GPL)
    - usata principalmente per le librerie di uso comune, consente l'uso di queste ultime in programmi proprietari
    - ogni modifica al codice della libreria deve comunque essere rilasciata sotto LGPL

# Licenze per il Software Libero (2)

---

- Software libero senza copyleft
  - BSD
    - licenza che consente libero uso, distribuzione, modifica del software
    - chiede che vengano riportate notizia del copyright e del contenuto della licenza
    - limita l'uso del nome dell'autore nella promozione dei prodotti derivati
    - non impone restrizioni sui prodotti derivati.
  - X11, Apache
- Per una classificazione più completa:
  - <http://www.gnu.org/licenses/license-list.html>

## Restrizioni al Software Libero

---

- Brevetti sull'hardware e mancanza di specifiche
- Brevetti sul software
  - possibilità di brevettare algoritmi o idee legate al software
    - GIF compresso con l'algoritmo LZW
  - protezione della “proprietà intellettuale”
- Leggi contro il “reverse engineering”
- Leggi “contro” la diffusione del software libero
  - legge 248/2000 (*bollino SIAE*)
- Librerie non libere
  - Motif e QT
- Documentazione non libera

## Disinformazione sul Software Libero

---

- Dovuta a varie cause
  - mancanza di cultura
  - pratiche di **FUD** (*Fear, Uncertainty and Doubt*)
- Sono nate molte leggende urbane
  - “Come fa il povero programmatore a campare?”
  - “Il software libero è per gli hacker” (=criminali)
  - “Il software libero è meno sicuro di quello proprietario”
  - “Il software libero è più lento”
  - “Il software libero è meno aggiornato”
- Il Manifesto GNU dimostra che sono *false*
  - <http://www.linux.it/GNU/manifesto.html>



## Cenni storici

---

- 1971: Richard Stallman entra al MIT
- 1984: Progetto GNU (GNU's Not Unix)
  - realizzare un **sistema operativo** libero per sviluppare ed utilizzare software libero
  - <http://www.gnu.org>
- 1985: Free Software Foundation
  - <http://www.fsf.org>
- Per una trattazione più completa:
  - <http://www.gnu.org/gnu/thegnuproject.it.html>



## Riferimenti

---

- Associazione Software Libero
  - <http://www.softwarelibero.it>
- Siti in italiano
  - [www.linux.it/GNU/](http://www.linux.it/GNU/)
    - Manifesto GNU
    - Trappola nel Cyberspazio
  - [www.apogeonline.it](http://www.apogeonline.it)
    - La cattedrale e il bazaar
  - [www.swlibero.org](http://www.swlibero.org)
    - Appunti di informatica libera
- Filosofia GNU (anche in italiano)
  - <http://www.gnu.org/philosophy/>

---

# Software Libero e Linux

Linux Day 6  
23 novembre 2002

---

## Linee evolutive

*Thompson  
Ritchie  
Kernighan  
Tanenbaum*

71: Unix



“Dialetti”



SV/BSD/Minix

*Stallman*

84: FSF



89: GNU



*Torvalds*

91: Linux



*Moore*

81: PC Intel

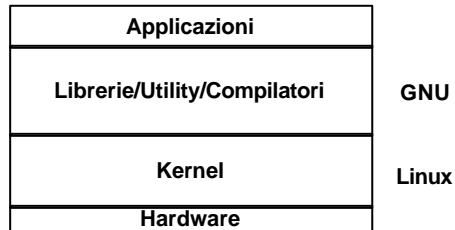


i386

**GNU/Linux**

# GNU/Linux

- 1991: Linus Torvalds sviluppa il kernel 0.0.2
- 1994: Kernel 1.0
- 2000: Kernel 2.4
- Architettura:
  - kernel
  - parte GNU
  - applicazioni
- Meriti di Torvalds?
- Per maggiori informazioni:
  - <http://www.gnu.org/gnu/linux-and-gnu.it.html>
  - <http://www.linux.{org|com|it}>



## Il “caso” Linux

- Vantaggi
  - sistema operativo libero molto diffuso
  - strumento a disposizione per utenti e sviluppatori
  - incremento di utenti di software libero
- Svantaggi
  - non tutte le distribuzioni sono “libere”
    - alcuni programmi proprietari
  - non tutto il software Linux è “libero”
  - eccessiva attenzione sul “*prodotto*” e non sull’*idea*

## Altri casi di Software Libero

---

- TeX
- XFree86
- GNOME
- Apache

## Successi del Software Libero

---

- GNU/Linux
  - 25% dei server
  - *the only non-Microsoft OS that is gaining market share*
- Apache
  - 60% dei server web
- TCP/IP
  - standard “de facto”

# Riferimenti

---

- Documenti
  - Manuali/guide delle distribuzioni, manuale in linea
  - Appunti di informatica libera (*ex appunti di Linux*)
    - <http://a2.swlibero.org/>
  - Linux Facile
    - <http://www.linuxfacile.org/>
  - HowTo, LDP
    - <http://www.tldp.org/>
- Riviste
- Siti
- Comunità Linux
  - LUG (lug@lugbs.linux.it)

---

## Software Libero e Didattica

Linux Day 6  
23 novembre 2002

---

## Vantaggi per gli studenti

---

- Libertà nell'uso del software
  - liberamente distribuibile
- Disponibilità di una “comunità”
  - scambio di informazioni, esperienze e programmi
- Riusabilità del proprio lavoro
  - sia per il software sviluppato che per i dati
- Possibilità di accrescere il proprio know how
  - il codice sorgente è il “know how” di un programma
  - sono richieste maggiori conoscenze tecniche
- Maggiori possibilità sul mondo del lavoro
  - più “flessibilità” mentale nell'uso del software libero
  - molte tecnologie “svaniscono” col tempo
  - larga diffusione del software libero
    - Unix ed è utilizzato su più del 40% dei server

## Vantaggi per i docenti

---

- Libertà nell'uso del software
  - liberamente distribuibile
- Disponibilità di una “comunità”
  - scambio di informazioni, esperienze e programmi
- Riusabilità del proprio lavoro
  - sia per il software sviluppato che per i dati
- Minor obsolescenza del proprio know how
  - Unix e C, ad esempio, esistono da 30 anni

## Vantaggi per la scuola

---

- Libertà nell'uso del software
  - liberamente distribuibile
- Maggior controllo sul software
  - verificabilità di sistemi di sicurezza
- Protezione degli investimenti
  - investimento non in beni, ma in persone e/o know how
    - non è detto che vi sia un risparmio immediato
  - maggior indipendenza da evoluzioni tecnologiche
    - minor obsolescenza dell'hardware
  - maggior indipendenza dei fornitori di software
  - investimento a “lungo termine”

## Altri vantaggi

---

- La libertà dei dati
  - il formato dei dati è **noto**
    - possibilità di leggerli anche senza il programma
  - programmi proprietari (potrebbero) gestire in modo “non noto” i propri dati
    - grave se applicato ai dati personali
- Sicurezza “verificabile” del software
- Qualità “verificabile” del software
- Maggior rispetto della propria privacy
- Crescita culturale per le nuove generazioni

# Contribuire al Software Libero

---

- Perché contribuire?
  - vantaggi diretti ed indiretti
  - vantaggi personali e vantaggi alla comunità
- Come contribuire?
  - sviluppare software libero
  - migliorare software libero
  - testare software libero
  - diffondere software libero
  - utilizzare software libero
  - aumentare l'informazione e la cultura informatica

# Riferimenti

---

- Progetto “Software Libero nella Scuola - linuxdidattica”
  - <http://linuxdidattica.org>
    - “fusione” dei progetti “Software Libero nella Scuola” e “Linux nella didattica”
- Software libero utilizzabile nella didattica
  - <http://linuxbox.interfree.it/educazione.html>
- Pieghevole da distribuire nelle scuole
  - <http://linuxdidattica.org/docs/pieghevole/>