



Fondamenti di informatica

(Prof. Nicola Orio)

Computer

Corso di laurea in
Storia e tutela dei beni artistici e musicali



Hardware e Software – 1

Definizione di hardware

- ✓ Insieme dei dispositivi meccanici, magnetici, elettrici ed elettronici che compongono un calcolatore
 - Significa in generale “ferraglia”

Definizione di software:

- ✓ Insieme delle istruzioni e dei programmi usati per determinare le operazioni di un calcolatore
 - Creato sulla contrapposizione tra “hard” e “soft”
 - Il suffisso “-ware” è usato per indicare gruppi di oggetti di un tipo determinato (es. glassware, silverware)



Hardware e software – 2

Le prestazioni di un calcolatore dipendono dall'insieme di hardware e software

- ✓ Un calcolatore molto potente può essere poco sfruttato da software scadente
- ✓ Un software ha bisogno di un hardware adeguato per poter funzionare correttamente
 - Requisiti minimi dell'hardware per un software

L'hardware è sempre più potente e consente di sviluppare software sempre più complesso



Altri -ware

Il suffisso -ware è usato anche per indicare

- ✓ Shareware (da to share=condividere)
 - Software dimostrativo, utilizzabili per un periodo di prova o con funzionalità ridotte
 - Scelta di marketing, si parla anche di demoware
- ✓ Freeware (da free=gratis)
 - Software gratuito che non può essere ridistribuito
 - Quando contiene pubblicità si parla di adware
- ✓ Malware (da malicious=malevolo)
 - Software scritto per danneggiare gli utenti
 - Quando spia i dati privati si parla di spyware

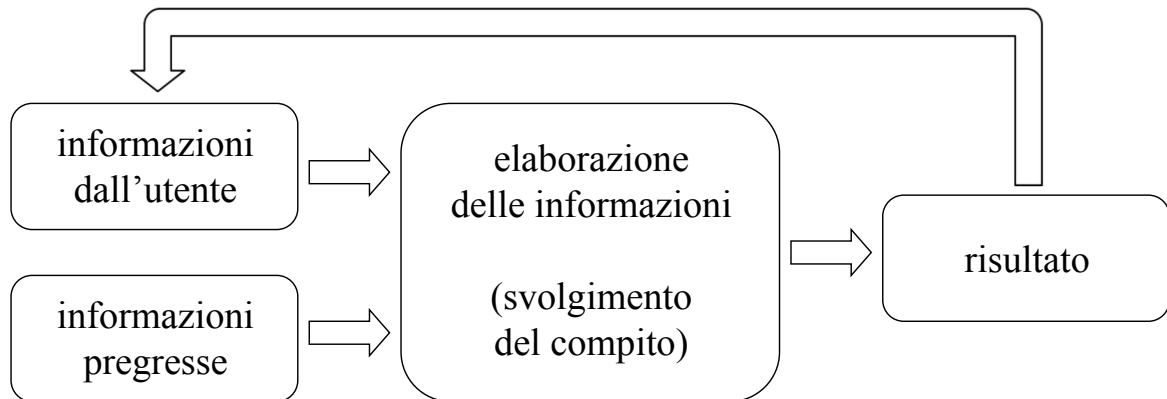


Compiti di un computer – 1

Il computer può svolgere compiti molto diversi

– Scrivere, calcolare, comunicare, svagarsi

I compiti possono essere schematizzati così:



Compiti di un computer – 2

Per formalizzare un compito ci si chiede

- ✓ Quali sono i dati iniziali e gli strumenti necessari?
- ✓ Come e da chi acquisire ulteriori informazioni?
- ✓ Che operazioni fare? In che ordine?
- ✓ Come presentare i risultati?

Tra queste domande, la più complessa è la terza

- ✓ Si devono formalizzare le singole attività
 - Possono essere correlate tra loro
 - L'ordine in cui vengono fatte è importante
 - Può essere necessario aggiustare le scelte



Dispositivi necessari

Per svolgere un compito servono

- ✓ Un dispositivo che esegua le operazioni
- ✓ Un posto dove memorizzare i dati
 - L'elenco di operazioni da eseguire
 - I dati che servono per l'operazione corrente
 - I dati che servono per tutto il compito
- ✓ Dispositivi per comunicare con l'esterno
 - Acquisire informazioni dal mondo esterno
 - Riportare i risultati o chiedere nuovi dati



Prestazioni

I dispositivi influenzano l'efficienza = rapidità

- ✓ Svolgere velocemente le singole operazioni
- ✓ Accedere molto rapidamente ai dati essenziali
 - Problemi se non c'è abbastanza spazio
- ✓ Accedere abbastanza rapidamente agli altri dati
 - Si assume uno spazio comunque sufficiente
- ✓ Comunicare rapidamente con il mondo esterno
 - Acquisire dati dall'utente non è un problema
 - Fornire dati all'esterno richiede efficienza



Architettura hardware

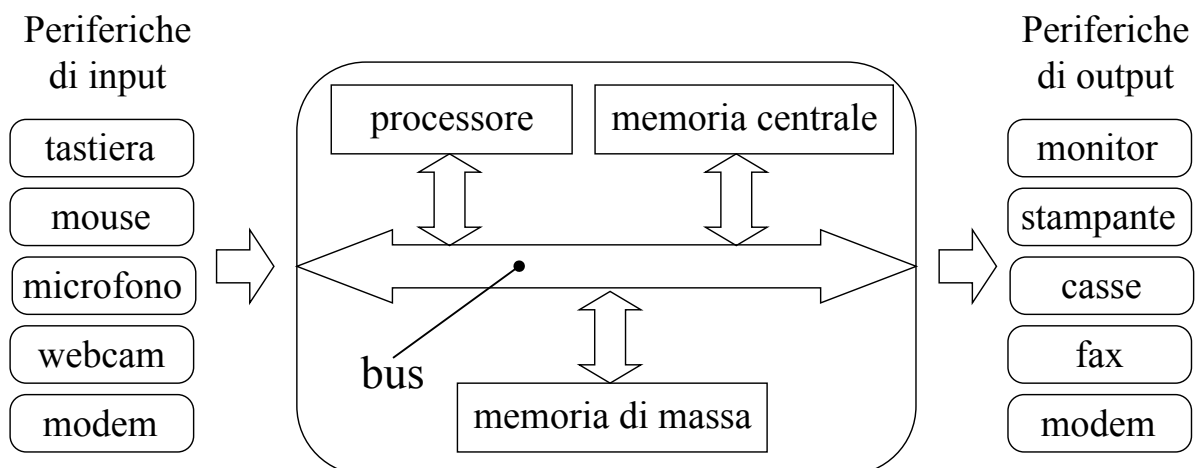
Gli elementi essenziali di un computer sono

- ✓ Processore, che esegue le operazioni
- ✓ Memoria veloce, per i dati di immediato utilizzo
 - Dati immediatamente necessari al processore
 - Istruzioni per la loro elaborazione
- ✓ Memoria capace, per altri dati
 - Dati utili in altre fasi dell'elaborazione
 - Altri programmi e risultati di elaborazioni precedenti
- ✓ Periferiche per la comunicazione
 - Di ingresso o input
 - Di uscita o output



Il computer

I computer attuali hanno architettura basata sulla macchina di Von Neumann (1946)





Processore – 1

Il processore svolge tutte le attività di un computer

- ✓ Viene spesso chiamato CPU (Central Processing Unit)
 - Un altro termine comune è microprocessore

Le operazioni sono elementari

- ✓ Calcoli aritmetici
- ✓ Confronto tra dati e scelte in base al risultato
- ✓ Lettura e scrittura dei dati in memoria
- ✓ Trasmissione dei dati da e verso l'esterno

Le operazioni sono organizzate in programmi complessi

- Rappresentati da sequenze finite di bit



Processore – 2

Il processore esegue le operazioni a cadenza prefissata

- ✓ Un segnale (clock) sincronizza il processore

Viene definito frequenza di clock il numero di operazioni che il processore esegue in un secondo

- Clock di 1GHz = 1 miliardo di operazioni al secondo

Importanza della frequenza di clock

- ✓ Influisce sostanzialmente sulla potenza di un computer
 - Più la frequenza di clock è elevata più rapido è il processore a eseguire i compiti/programmi
 - Le operazioni possono richiedere più di un ciclo di clock



Processore – 3

Il processore svolge molti compiti insieme

- ✓ Salta rapidamente da un compito all'altro
- ✓ L'utente percepisce più attività contemporanee

Ci possono essere più processori in una CPU

- ✓ Architetture multi-core
 - Tipicamente dual-core o quad-core ma anche oltre
- ✓ Molto efficienti quando
 - Il computer esegue più di un compito
 - Il compito è parallelizzabile



Memoria centrale – 1

La memoria centrale contiene tutti i dati necessari allo svolgimento di un compito

- ✓ L'elenco delle operazioni da svolgere
- ✓ I dati da elaborare acquisiti dall'esterno
- ✓ I risultati temporanei delle elaborazioni
- ✓ I dati elaborati da restituire verso l'esterno

I dati sono rappresentati in celle da uno o più byte

- ✓ Il processore scrive o legge nelle celle indicandone l'indirizzo (un numero)



Memoria centrale – 2

Tempo di accesso

- ✓ Tempo necessario per scrivere o leggere il contenuto di una locazione di memoria
- ✓ Caratteristica importante per le prestazioni
 - Il processore deve attendere mentre la memoria centrale legge o scrive i valori
 - Idealmente dovrebbe essere necessario al massimo un ciclo di clock
- ✓ Influisce sul costo della memoria
- ✓ Impone dei limiti nella tecnologia realizzativa



Memoria centrale – 3

Indicata con la sigla RAM (Random Access Memory)

Caratteristiche della RAM

- ✓ Deve essere estremamente veloce
 - Quindi è costosa
 - Con la tecnologia attuale perde le informazioni quando il calcolatore viene spento

Le dimensioni influiscono sulle prestazioni

- ✓ Quantità di dati accessibili dal processore
 - Attualmente nell'ordine dei gigabyte



Memoria di massa – 1

Rispetto alla memoria centrale, è

- ✓ Più economica
 - Quindi a parità di costo più capiente
- ✓ Più lenta
 - Il prezzo da pagare per l'economicità
- ✓ Permanente
 - Conserva i dati anche a computer spento
 - Può conservarli anche per molto tempo
 - Rischio comunque di perdita
 - Necessarie strategie di conservazione/replicazione



Memoria di massa – 2

Comprende diverse tecnologie

- ✓ Magnetica: dischi (in particolare hard-disk) e nastri
- ✓ Ottica: CD-Rom, DVD-Rom
- ✓ Memorie flash: chiavi USB, dischi a stato solido

Il processore non vi accede direttamente

- ✓ I dati devono essere caricati in RAM
- ✓ Rispetto alla memoria di massa le prestazioni
 - Dipendono dalla velocità di accesso
 - Non dipendono dalla capacità
 - Finché c'è spazio sufficiente non ci sono differenze di prestazioni



Memoria di massa – 3

Il componente principale è l'hard-disk (HD)

- ✓ Definito anche disco fisso
 - In alternativa ai floppy disk che erano rimovibili

L'HD è legato alla percezione di efficienza

- ✓ Tempo che l'utente deve attendere per
 - Avviare un programma
 - Caricare un file
- ✓ I dischi a stato solido (SSD) sono più veloci dei dischi magnetici tradizionali
 - Attenzione: Il numero di scritture è limitato



Periferiche – 1

Le periferiche di input (ingresso)

- ✓ Acquisiscono le informazioni dal mondo esterno
 - Gestì, immagini, suoni, segnali elettrici
 - Trasformati in sequenze di bit
 - Dati in forma trattabile dal computer

Le periferiche output (uscita)

- ✓ Forniscono informazioni al mondo esterno
 - Sequenze di bit
 - Trasformate in immagini, suoni, segnali elettrici
 - Dati in forma utilizzabile dal mondo esterno



Periferiche – 2

Il computer può diventare

- ✓ Multimediale
 - Sono utilizzati media diversi (testo, immagini)
- ✓ Multimodale
 - Sono utilizzate sensi diversi (vista, udito)

Le periferiche servono per comunicare

- ✓ Con l'utente
- ✓ Con apparecchiature esterne
- ✓ Con altri computer in rete



Rapporto tra CPU e RAM

Il processore, ad ogni passo:

- ✓ Richiede alla RAM qual è la prossima istruzione da eseguire
 - La RAM contiene tutti i programmi in esecuzione
- ✓ Esegue l'istruzione corrispondente
 - Se questa necessita di un dato, lo legge dalla RAM
 - Se questa produce un dato, lo scrive nella RAM
 - La RAM contiene tutti i dati utilizzati dalla CPU

Normalmente i dati e i programmi che li usano sono in due posizioni diverse della RAM

- ✓ Rischio di interpretare un dato come fosse un'istruzione



CACHE

Problema

- ✓ Una RAM molto veloce è molto costosa

Soluzione

- ✓ I dati che servono alla CPU spesso sono vicini
 - Le istruzioni dei programmi
 - I singoli elementi di un insieme di dati
- ✓ I dati più probabilmente utili alla CPU sono memorizzati in una memoria velocissima
 - Memoria cache, contenuta direttamente nella CPU
- Sono di solito copie di dati già usati o dati in locazioni vicine



SWAP

Problema

- ✓ La RAM può non riuscire a contenere tutti i dati necessari alla CPU

Soluzione

- ✓ Alla CPU non servono tutti i dati in ogni istante
- ✓ I dati che non servono sono temporaneamente salvati nell'HD (operazione denominata SWAP)
 - Se si nota un'attività eccessiva dell'HD può essere che la RAM sia sottodimensionata
 - Lo SWAP ha funzioni simili alla memoria cache



RAID

Problema

- ✓ Gli HD sono lenti e possono rompersi

Soluzione

- ✓ Utilizzare più HD, replicando i dati
 - Configurazioni RAID = Redundant Array of Independent Disks
- La CPU può trasferire i dati da più HD contemporaneamente
 - Aumenta la velocità di accesso complessiva
- Se un HD si rompe i dati si reperiscono da un altro
 - Basso rischio che si rompano due HD insieme



ROM

Problema

- ✓ La CPU legge dalla RAM le istruzioni da eseguire, ma all'accensione la RAM è vuota

Soluzione

- ✓ Esiste una memoria di sola lettura, che mantiene le informazioni a computer spento
 - ROM = Read Only Memory
- ✓ All'accensione la CPU legge le istruzioni dalla ROM
 - Grazie a queste può accedere ai programmi nell'HD
 - La ROM non è l'HD



Buffer

Problema

- ✓ La comunicazione con le periferiche è asincrona
 - Non si sa quando l'utente clicca con il mouse
 - Non si sa quando la stampante ha finito una pagina

Soluzione

- ✓ Memoria di transito dove accodare i dati in trasferimento
 - Memoria buffer o memoria tampone
- ✓ Le periferiche leggono/scrivono nel buffer
 - La CPU non è rallentata dall'interazione
 - Non c'è rischio di perdita di dati



Driver

Problema

- ✓ Il computer non conosce tutti i modelli di periferica

Soluzione

- ✓ Per ogni periferica c'è un piccolo programma che descrive come utilizzarla
 - Tecnicamente chiamato driver
 - Spesso il driver è contenuto nella periferica stessa
 - Utilizzo di memorie flash
 - Altrimenti bisogna reperirlo (CD-Rom, Internet)
 - Problemi con le periferiche per sistemi commerciali



Futuro dell'hardware

Legge di Moore (1965)

- ✓ La potenza di calcolo raddoppia ogni circa due anni
 - Valida fino ad oggi
 - La miniaturizzazione sta raggiungendo i limiti atomici

Computer ottici?

- ✓ Fotoni invece di elettroni

Computer quantici?

- ✓ Qubit (quantum bit) invece di bit
 - Un qubit può valere 0 e 1 contemporaneamente



Programmazione

Scrivere software è un'attività complessa

- ✓ Formalizzare i compiti da svolgere
 - Si parla di scrittura di algoritmi
- ✓ Prevedere e gestire tutte le situazioni
 - Situazioni impreviste possono bloccare l'algoritmo
- ✓ Difficile fare test completi

Ci si appoggia a software già scritto

- ✓ Algoritmi complessi basati su algoritmi più semplici
 - Bisogna tenere traccia dei diversi componenti



Sistemi informativi

Ci sono diversi livelli di sistema informatico

- ✓ Dispositivo mobile
 - Un solo utente, controllo limitato
- ✓ Personal computer
 - Più utenti (non insieme), controllo quasi totale
- ✓ Server
 - Molti utenti insieme, un amministratore controlla
- ✓ Cloud
 - Molti computer, molti utenti, più amministratori



Tipologie di software

Per il computer i programmi servono

- ✓ A farlo funzionare
 - Senza questo software l'hardware è inutile
 - Essendo così importante è già installato all'acquisto
- ✓ A specializzarlo per svolgere compiti particolari
 - Renderlo uno strumento di lavoro/svago
 - Alcuni programmi sono già installati, per comodità
 - Ogni utente aggiunge gli altri programmi a sua scelta

Per il processore non c'è differenza

- Sono in tutti i casi delle sequenze di operazioni



Sistema operativo – 1

Consente l'uso del computer

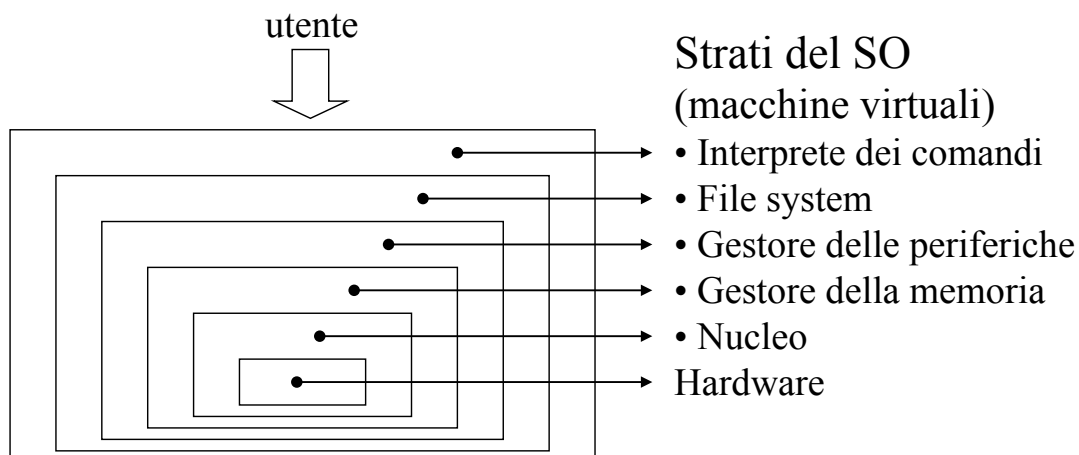
- ✓ Insieme di programmi che
 - Controlla e sincronizza i programmi attivi
 - Gestisce la memoria centrale
 - Consente la comunicazione con le periferiche
 - Gestisce i dati in memoria di massa
 - Interpreta le istruzioni dell'utente
- ✓ Gli errori si riflettono su tutto il sistema
 - Pochi sistemi operativi in commercio



Sistema operativo – 2

Organizzato in più strati che

- ✓ Implementano una funzionalità
- ✓ Comunicano prevalentemente con gli strati vicini





Interprete dei comandi

Strato più esterno per comunicare con l'utente

- ✓ Interazione testuale
 - Istruzioni in forma di comandi utilizzando la tastiera
 - Bisogna ricordare nome e sintassi dei comandi
 - Si possono eseguire operazioni complesse
- ✓ Interazione grafica
 - Funzioni rappresentate graficamente da icone
 - Selezione con interfaccia di puntamento
 - Bisogna selezionare ogni volta gli oggetti e i comandi da dare
 - Invece di ricordare i comandi si devono riconoscere le icone



File System – 1

Si occupa della gestione dei file, che

- ✓ Sono porzioni della memoria di massa, con un nome
 - Di solito diviso in due parti
 - Nome vero e proprio: sequenza di caratteri
 - Estensione: un punto seguito da uno o più caratteri
 - Creato dall'utente o dai programmi stessi
 - Appiglio mnemonico per ipotizzare il contenuto del file
 - La cui estensione indica
 - Come sono organizzati i dati contenuti nei file
 - Il tipo di programma in grado di interpretare il file



File System – 2

I file risiedono in memoria di massa

- ✓ La struttura fisica del supporto è trasparente
- ✓ Si possono creare dei supporti virtuali
 - Non corrispondono a degli effettivi supporti fisici
- ✓ Lo spazio è gestito automaticamente
 - Ricerca di spazio libero per nuovi file
 - Cancellazione rapida

Sistemi multiutente

- ✓ Gestione del controllo degli accessi



File System – 3

Fornisce dei metodi di uso generale

- ✓ Leggono, scrivono, cancellano, aggiornano i file
- ✓ Possono essere invocati da
 - Utente tramite l'interprete dei comandi
 - Applicazioni che gestiscono documenti
 - Anche altri strati del sistema operativo
- ✓ Accede ai file specificandone il nome
 - Il nome di un file dovrebbe essere unico
 - scomodo per gli utenti
 - Problemi con tanti file diversi e di diversa provenienza



Directory – 1

Definizione di directory

- ✓ Struttura virtuale gestita dal file system che può
 - Essere vuota
 - Contenere uno o più file
 - Possibili limitazioni al numero di file
 - Contenere una o più directory
 - Possibili limitazioni alla profondità di sub-directory

Ha la funzione di organizzare i file

- ✓ Utile per l'utente e per molte applicazioni



Directory – 2

Struttura ad albero rovesciato

- Directory radice (root)
 - Non è contenuta in nessun'altra directory
- Altre directory sono i rami
- I file sono le foglie
- ✓ Unico percorso per raggiungere il file
 - Chiamato pathname
 - Pathname = posizione + nome

Esempio (stile web): /user/orio/documenti/unipd/software.pptx

- File con posizione diversa possono avere lo stesso nome



Installazione

Per installare nuovo software bisogna

- ✓ Copiare il codice in memoria di massa
 - Creare nuovi file e directory utili al software
 - Creare le icone associate
- ✓ Notificare la presenza al sistema operativo
 - Il software può avere funzionalità utili anche ad altri software

Spesso gestita da programmi detti wizard

- ✓ Attenzione a cosa viene installato
- ✓ Alcuni programmi sono venduti preinstallati nel SO



Avvio

Le applicazioni vengono avviate da

- ✓ Utente tramite l'interprete dei comandi
- ✓ Altre applicazioni (anche dal sistema operativo)

All'avvio

- Il sistema operativo
 - Carica in memoria centrale il file dell'applicazione
- La CPU
 - Esegue le istruzioni elementari contenute nel file
 - Non fa nessun controllo sulla correttezza del software
- L'utente
 - Deve fidarsi che tutto vada a buon fine e che non sia malware



Disinstallazione

Quando un'applicazione non interessa più

- ✓ Cancellare il software non è sufficiente
- ✓ Bisogna notificare la rimozione al sistema operativo
 - Vengono cancellati anche i file di utilità nel sistema

Operazione non facile per utenti inesperti

- ✓ Molti sistemi operativi hanno strumenti adeguati
 - Gestione applicazioni in Windows
 - Cancellazione più facile per Mac OS X, Android, iOS
- ✓ Cancellazione guidata per alcuni programmi
 - Ai programmatori conviene sia macchinoso



Codice macchina

Il processore esegue operazioni molto semplici

- ✓ I programmi sono costituiti da milioni istruzioni elementari
 - Il processore è rapidissimo ad eseguirle
 - L'utente non ha la percezione di quanto siano elementari

Problema: troppe istruzioni, troppo semplici

- ✓ E' quasi impossibile capire cosa farà un programma leggendo il codice macchina
- ✓ I programmatori scrivono in codice macchina in casi rarissimi, quando l'efficienza è fondamentale



Codice sorgente

I programmatori usano linguaggi ad alto livello

- ✓ Conversione automatica in linguaggio macchina
 - Eseguita da software detti compilatori e interpreti
 - Da codice sorgente (linguaggio del programmatore)
 - A codice macchina (linguaggio del processore)
 - Il processo non è invertibile
 - Dal codice macchina è quasi possibile risalire al codice sorgente
 - Modifiche, correzioni, aggiornamenti sono fatte sul codice sorgente
 - Viene generato e distribuito un nuovo codice macchina



Open source – 1

Con l'open source si rilascia il codice sorgente

- ✓ Per funzionare basta il codice macchina
- ✓ Se altri programmatori accedono al codice sorgente
 - Possono copiarlo e/o rivenderlo
 - I propri “trucchi” diventano noti
 - Le imperfezioni o le parti mancanti sono visibili
 - Possono aggiornarlo per conto proprio
 - Non ci si deve rivolgere al programmatore iniziale
 - Perdita del controllo del proprio software
 - Le grandi software-house non amano l'open source



Open source – 2

Motivazioni a favore

- ✓ Per i programmatori
 - Collaborazione per software efficiente e sicuro
 - Molte parti già disponibili da adattare alle proprie esigenze
 - Alternativa al potere delle grandi software house
- ✓ Per gli utenti
 - Sicurezza e affidabilità
 - Si può leggere cosa fa un programma
 - Indipendenza
 - Si può chiedere a terzi di fare modifiche
 - Personalizzazione



Creative Commons

Concetto di open source applicato ai documenti

- ✓ Applicato a tutti i media
 - Musica, fotografie, video, testi
- ✓ Diversi gradi di licenza
 - Dall'utilizzo in qualsiasi forma
 - Anche commerciale, anche di parti
 - Al solo ascolto
 - Solo integrare e per uso personale

Concetti mutuati da open source e shareware



Trusted computing

Nato per evitare intercettazioni e manomissioni

- ✓ Hardware e software appositi
 - Strumenti principali
 - Cifratura del software (endorsement key)
 - Controllo periferiche (secure I/O)
 - Memorie separate (memory curtaining)
 - Memoria bloccata (sealed storage)
 - Controllo remoto dello stato (remote attestation)
- ✓ Poco noto ma supportato dalle multinazionali
 - Diminuzione del controllo da parte degli utenti
 - Difficile bilanciamento tra sicurezza e privacy