

Corso Sistemi Operativi

Ing. Pierfrancesco Bellini
pierfrancesco.bellini@unifi.it
Laboratorio DISIT
Dip. Ingegneria dell'Informazione
Via S. Marta, 3

Programma a.a. 2015/16

- Introduzione
 - Introduzione ai sistemi operativi
 - Struttura dei sistemi operativi
- Le basi del linguaggio Java
- Gestione dei processi
 - I processi
 - I thread
 - Scheduling della CPU
 - Sincronizzazione tra processi
 - Stallo
- Gestione della memoria
 - Gestione della memoria centrale
 - Memoria virtuale

Libri di testo principali

A. Silberschaz, P. Galvin, G. Gagne,
Sistemi Operativi: Concetti ed esempi,
nona edizione, Pearson Addison-Wesley.

K. Arnold, J. Gosling, D. Holmes
Java manuale ufficiale,
Pearson Addison-Wesley

Altro materiale su:

<http://www.disit.org/sistemi-operativi>

Sistemi Operativi A.A 2015/2016

Esame

- Esame scritto e orale
- Due prove in itinere sostituiscono lo scritto
- Non è previsto il salto appello
- Uno scritto sufficiente è valido per i due appelli successivi
- All'orale lo studente deve portare una implementazione funzionante degli esercizi della prova scritta.

Sistemi Operativi A.A 2015/2016

Sistemi Operativi

Introduzione

Cosa è un Sistema Operativo?

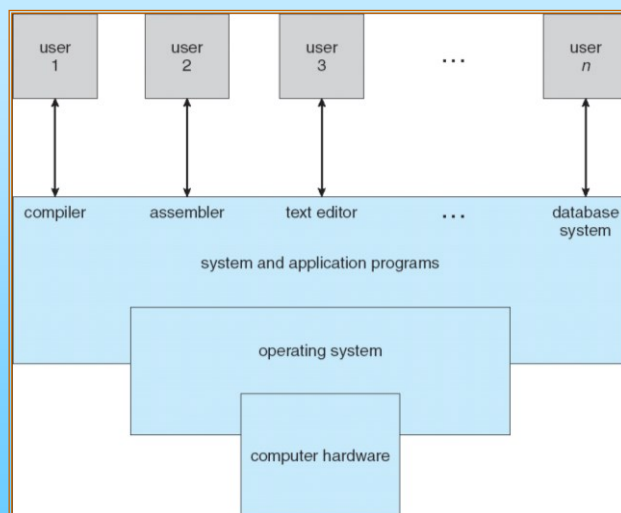
- Un programma che agisce come intermediario tra un utente e l'hardware dell'elaboratore.
- Obiettivi dei Sistemi Operativi:
 - Eseguire i programmi degli utenti e facilitare la risoluzione dei problemi degli utenti.
 - Rendere conveniente l'uso dei calcolatori.
 - Usare l' hardware del calcolatore in modo efficiente.

Struttura di un Calcolatore

- Un Calcolatore può essere diviso in 4 componenti
 - **Hardware** – fornisce componenti di base
 - ▶ CPU, memoria, dispositivi I/O
 - **Sistema Operativo**
 - ▶ Controlla e coordina l'uso dell'hardware tra le varie applicazioni e gli utenti
 - **Programmi Applicativi** – definiscono i modi in cui le risorse del sistema sono usate per risolvere i problemi computazionali degli utenti
 - ▶ Word processors, compilatori, web browsers, database systems, video games
 - **Utenti**
 - ▶ Persone, macchine, altri calcolatori

Sistemi Operativi A.A 2015/2016

Quattro componenti di un Calcolatore



Sistemi Operativi A.A 2015/2016

Definizione di Sistema Operativo

- Il Sistema Operativo è un **allocatore di risorse**
 - Gestisce tutte le risorse
 - Decide tra richieste in conflitto per un uso efficiente e giusto delle risorse disponibili.
- Il Sistema Operativo è un **programma di controllo**
 - Controlla l'esecuzione dei programmi per prevenire errori e un uso improprio del calcolatore
- Non esiste una definizione universalmente accettata
- "Il programma sempre in esecuzione sul calcolatore" e' il **kernel (nucleo)**. Ogni altra cosa o e' un programma di sistema (venduto con il sistema operativo) o e' un programma applicativo

Sistemi Operativi A.A 2015/2016

Avvio del calcolatore

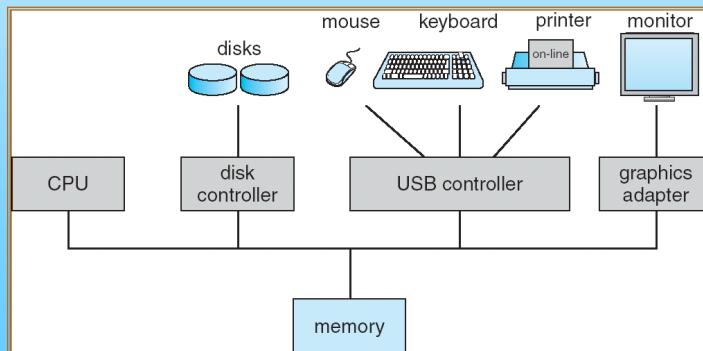
- Il **programma di bootstrap** e' caricato all'accensione o al reboot
 - Tipicamente memorizzato in ROM o EEPROM, noto come **firmware**
 - Inizializza tutti gli aspetti del sistema
 - Carica il kernel del sistema operativo e inizia la sua esecuzione

Sistemi Operativi A.A 2015/2016

Organizzazione di un Calcolatore

■ Nel Calcolatore:

- Una o più CPU e controllori di dispositivi connessi attraverso un bus comune che fornisce accesso a una memoria condivisa
- Esecuzione concorrente delle CPU e dispositivi che competono per l'uso della memoria (memory cycles)



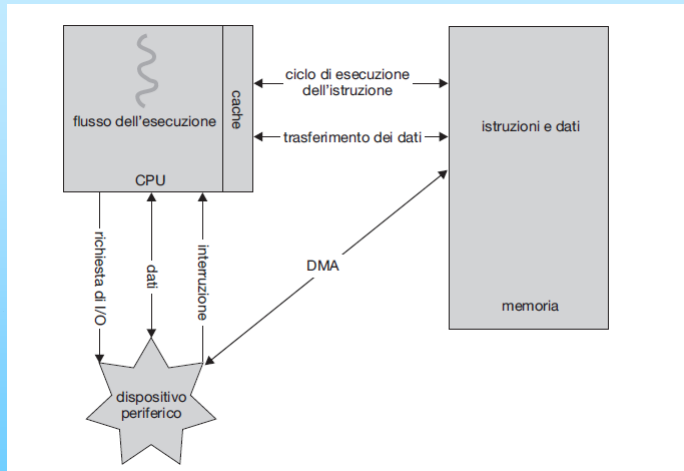
Sistemi Operativi A.A. 2015/2016

Calcolatore

- Dispositivi I/O e la CPU possono operare in modo concorrente.
- Ogni controllore di dispositivo si occupa di un tipo di dispositivo.
- Ogni controllore di dispositivo ha un suo buffer locale.
- CPU sposta dati da/a memoria a/da buffer locali
- I/O e' dal dispositivo al buffer locale del controllore.
- Il controllore di dispositivi informa la CPU che ha finito la sua operazione causando una interruzione (*interrupt*).

Sistemi Operativi A.A. 2015/2016

Calcolatore



Sistemi Operativi A.A 2015/2016

Funzioni comuni delle Interruzioni

- L'interruzione trasferisce il controllo alla routine di gestione della interruzione attraverso il **vettore delle interruzioni** che contiene gli indirizzi di tutte le routine di servizio.
- L'interruzione deve salvare l'indirizzo dell'istruzione interrotta.
- Le interruzioni vengono *disabilitate* quando una interruzione viene gestita per impedire la *perdita di interruzioni*.
- Una *trap* è una interruzione generata dal software causata o da un errore o da una richiesta dell'utente.
- I sistemi operativi sono guidati dalle interruzioni (*interrupt driven*).

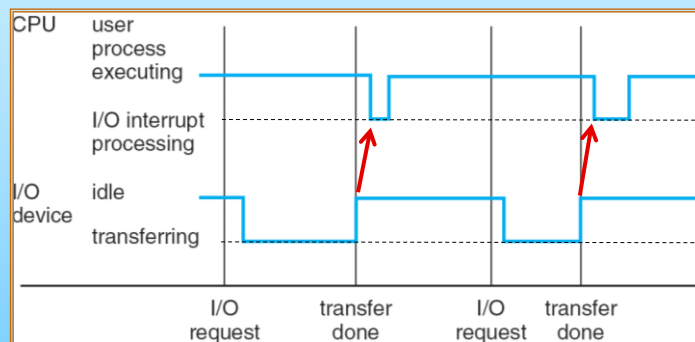
Sistemi Operativi A.A 2015/2016

Gestione delle Interruzioni

- Il Sistema Operativo preserva lo stato della CPU memorizzando il contenuto dei registri e del program counter.
- Determina che interruzione e' avvenuta:
 - *polling*
 - *vectored* interrupt system
- Segmenti di codice separati determinano che azioni intraprendere per ogni tipo di interrupt

Sistemi Operativi A.A 2015/2016

Timeline delle Interruzioni



Sistemi Operativi A.A 2015/2016

Architettura degli elaboratori

■ Sistemi monoprocessoore

- Possiedono una sola CPU per l'esecuzione dei programmi utente
- Possono avere altri processori che svolgono compiti specifici es. controller del disco (ma non sono gestiti dal SO)

■ Sistemi multiprocessoore

- Più di una unità di elaborazione strettamente connesse, condividono il bus, il clock e dispositivi di memorizzazione e periferici.
- Vantaggi:
 - ▶ **Maggiore produttività**
 - ▶ **Economia di scala**, condivisione periferiche
 - ▶ **Incremento della affidabilità**, un guasto non blocca il sistema, lo rallenta (fault-tolerant)

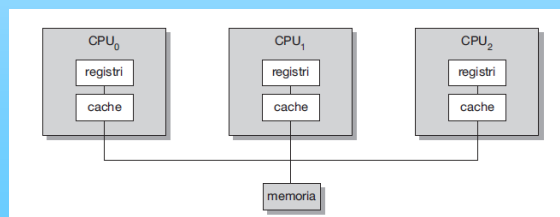
Sistemi Operativi A.A 2015/2016

Sistemi multiprocessoore

■ Di due tipi:

- **Multielaborazione asimmetrica (AMP)**, in cui ogni unità ha un compito specifico e una unità principale coordina e assegna il lavoro alle altre unità (soluzione poco frequente)
- **Multielaborazione simmetrica (SMP)**, in cui ogni processore è abilitato ad eseguire tutte le funzioni del sistema (soluzione più frequente)

- Tendenza recente di avere più **core** in uno stesso circuito integrato (risparmio energia)



Sistemi Operativi A.A 2015/2016

Struttura della memoria

- **Memoria primaria** – la sola unità di memorizzazione che la CPU può accedere direttamente.
- **Memoria secondaria** – estensione della memoria primaria che fornisce una grande capacità di memorizzazione non volatile.
- **Dischi magnetici** – piatti di metallo o vetro coperti da materiale magnetizzabile
 - La superficie del disco è divisa in *tracce* che sono suddivise in *settori*.
 - Il controller del disco determina l'interazione tra il dispositivo e il computer.

Sistemi Operativi A.A 2015/2016

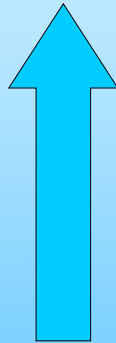
Gerarchia della Memoria

- Sistemi di memorizzazione organizzati in una gerarchia.
 - Velocità
 - Costo
 - Volatilità
- **Caching** – copiare informazioni in un sistema di memorizzazione più veloce; la memoria primaria può essere vista come *cache* per la memoria secondaria.

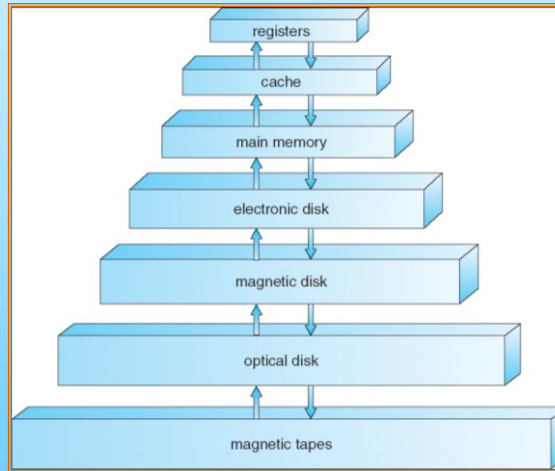
Sistemi Operativi A.A 2015/2016

Gerarchia dispositivi memorizzazione

+veloci, +costosi, volatili



-veloci, -costosi,
permanenti



Sistemi Operativi A.A 2015/2016

Caching

- Principio importante, utilizzato a vari livelli in un calcolatore (nell'hardware, sistema operativo, software)
- Informazione in uso viene copiata temporaneamente da dispositivo di memorizzazione lento a uno più veloce.
- Memoria veloce (cache) controllata prima per determinare se l'informazione è presente
 - Se lo è, l'informazione è usata direttamente dalla cache (veloce)
 - Altrimenti, dati copiati nella cache e usati da lì
- La cache è più piccola della memoria dietro la cache
 - Gestione della cache, importante problema di progettazione
 - Dimensione della cache e politica di rimpiazzamento

Sistemi Operativi A.A 2015/2016

Performance a vari livelli di memorizzazione

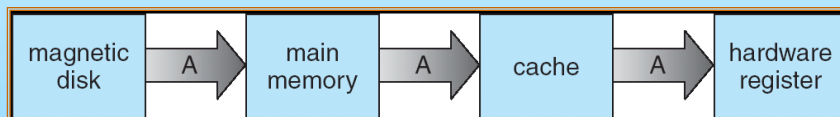
- Movimenti di dati tra livelli della gerarchia possono essere espliciti (es. caricamento dati da disco) o impliciti (es. cache processore)

Level	1	2	3	4
Name	registers	cache	main memory	disk storage
Typical size	< 1 KB	> 16 MB	> 16 GB	> 100 GB
Implementation technology	custom memory with multiple ports, CMOS	on-chip or off-chip CMOS SRAM	CMOS DRAM	magnetic disk
Access time (ns)	0.25 – 0.5	0.5 – 25	80 – 250	5,000.000
Bandwidth (MB/sec)	20,000 – 100,000	5000 – 10,000	1000 – 5000	20 – 150
Managed by	compiler	hardware	operating system	operating system
Backed by	cache	main memory	disk	CD or tape

Sistemi Operativi A.A 2015/2016

Migrazione di un intero A dal disco al registro

- In ambienti **multitasking** si deve stare attenti ad usare il valore più recente, indipendentemente di dove è memorizzato nella gerarchia delle memorie



- Sistemi **multiprocessore** devono fornire coerenza della cache in hardware in modo tale che tutte le CPU abbiano lo stesso valore nella loro cache.
- In ambienti **distribuiti** la situazione è ancora più complessa
 - Possono esistere molte copie del dato in calcolatori diversi

Sistemi Operativi A.A 2015/2016

Struttura di un Sistema Operativo

- **Multiprogrammazione** necessaria per efficienza
 - Un singolo utente non può tenere la CPU e i dispositivi I/O sempre occupati
 - La multiprogrammazione gestisce lavori (jobs) (codice e dati) in modo che la CPU abbia sempre qualcosa da fare...
 - Un sottoinsieme dei lavori presenti sul sistema è tenuta in memoria
 - Un lavoro viene selezionato ed eseguito attraverso il **job scheduling**
 - Quando deve aspettare (per I/O per esempio), OS commuta su un altro job.

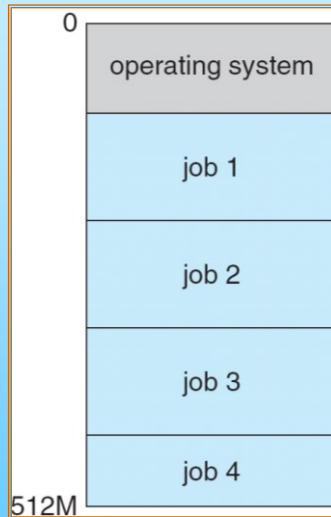
Sistemi Operativi A.A 2015/2016

Struttura di un Sistema Operativo

- **Timesharing (multitasking)** è la estensione dove la CPU cambia job così frequentemente che l'utente può interagire con il job mentre e' in esecuzione, dando vita all' **interactive computing**
 - **Tempo di risposta** dovrebbe essere < 1 secondo
 - Ogni utente ha almeno un programma in esecuzione in memoria \Rightarrow **processo**
 - Se molti job sono pronti all'esecuzione allo stesso tempo \Rightarrow **CPU scheduling**
 - Se i processi non stanno tutti in memoria, lo **swapping** li sposta in/fuori dalla memoria per essere eseguiti
 - La **memoria virtuale** permette l'esecuzione di processi anche non completamente in memoria

Sistemi Operativi A.A 2015/2016

Schema della memoria in un sistema multiprogrammato



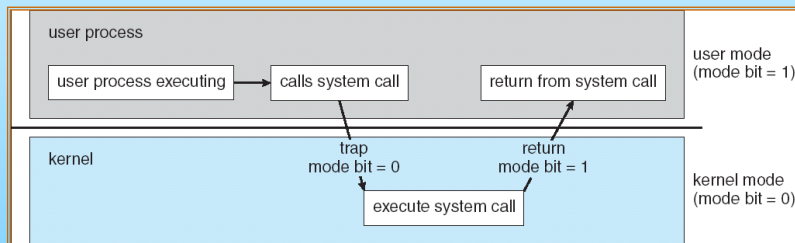
Sistemi Operativi A.A 2015/2016

Sistema Operativo

- Interruzioni guidate dall'hardware
- Un errore software o una richiesta crea una **eccezione** o una **trap**
 - Divisione per zero, richiesta di un servizio del SO
- Altri problemi dei processi includono: i loop infiniti, processi che si modificano a vicenda o modificano il sistema operativo
- **Dual-mode** permette al SO di proteggersi e proteggere altri componenti del sistema
 - **User mode e kernel mode**
 - **Bit di modo** fornito dall'hardware
 - ▶ Permette di distinguere quando il sistema sta eseguendo codice utente o codice del kernel
 - ▶ Alcune istruzioni designate come **privilegiate** sono eseguibili come in modo kernel
 - ▶ Le chiamate al sistema cambiano il modo a modo kernel e al termine della chiamata ritorna al modo utente

Sistemi Operativi A.A 2015/2016

Transizione da modo utente a modo kernel



Sistemi Operativi A.A 2015/2016

Timer

- Il SO usa un timer per prevenire i loop infiniti / processi che si impossessano delle risorse del calcolatore senza rilasciarle
 - Il timer hardware genera un interrupt ad intervalli regolari
 - Il Sistema Operativo decrementa un contatore
 - Quando il contatore raggiunge lo zero genera un interrupt
 - Il contatore viene impostato dal sistema operativo prima di schedare un processo, per riottenere il controllo e terminare il processo se occupa più tempo del previsto.

Sistemi Operativi A.A 2015/2016

Gestione Processi

- Un processo è un programma in esecuzione. E' una unità di lavoro nel sistema di calcolo. Un Programma è una **entità passiva**, un processo è una **entità attiva**.
- Un processo ha bisogno di risorse per svolgere il suo compito
 - CPU, memoria, I/O, files
 - Dati iniziali
- La terminazione di un processo implica la riappropriazione da parte del SO di ogni risorsa riusabile
- Un *processo single-thread* ha un **program counter** che specifica la locazione della prossima istruzione da eseguire
 - Il processo esegue le istruzioni sequenzialmente, una alla volta, fino al completamento
- Un *processo multi-thread* ha un program counter per thread
- Tipicamente il sistema ha molti processi in esecuzione su una o più CPU, alcuni processi sono degli utenti altri del sistema operativo
 - Concorrenza ottenuta facendo multiplexing della/e CPU tra i processi / threads

Sistemi Operativi A.A 2015/2016

Attività della Gestione dei Processi

Il Sistema Operativo è responsabile delle seguenti attività per la gestione dei processi:

- Creare e cancellare processi utente o del sistema operativo
- Sospendere e riprendere processi
- Fornire meccanismi per la sincronizzazione dei processi
- Fornire meccanismi per la comunicazione tra processi
- Fornire meccanismi per la gestione dello stallo (deadlock)

Sistemi Operativi A.A 2015/2016

Gestione della Memoria

- Tutti i dati sono in memoria prima e dopo l'esecuzione
- Tutte le istruzioni devono essere in memoria per poter essere eseguite
- La gestione della memoria determina cosa è in memoria e quando
 - Ottimizza l'utilizzo della CPU e la risposta del sistema
- Attività di gestione della memoria
 - Tiene traccia di quali parti della memoria sono utilizzate e da chi
 - Decide quali processi (o sue parti) e dati muovere in/out dalla memoria primaria
 - Alloca e disalloca spazi di memoria in base alle necessità

Sistemi Operativi A.A. 2015/2016

Gestione Memoria Secondaria

- Il Sistema Operativo fornisce una vista uniforme e logica della memoria secondaria
 - Astrae proprietà fisiche (dischi, tracce, settori) in unità logiche di memorizzazione, i **file**
 - Ogni medium è controllato da un dispositivo (i.e., disk drive, tape drive)
 - Si differenziano per velocità di accesso, capacità, velocità di trasferimento, metodo di accesso (sequenziale o diretto)
- Gestione del File-System
 - Files solitamente organizzati in cartelle/directories
 - Controllo degli accessi per determinare chi può usare cosa
 - Attività del SO includono
 - Creare e cancellare files e directories
 - Primitive per manipolare files e dirs
 - Mappare files in memoria secondaria
 - Backup files su supporti di memorizzazione stabili (non volatili)

Sistemi Operativi A.A. 2015/2016

Gestione Memorie di massa

- Solitamente i dischi sono usati per memorizzare dati che non entrano in memoria primaria o dati che devono essere memorizzati per lungo tempo.
- Una gestione oculata è fondamentale
- L'intera velocità del calcolatore è basata sul sottosistema di gestione dei dischi e sui suoi algoritmi di gestione
- Attività del SO
 - Gestione dello spazio libero
 - Allocazione
 - Scheduling del disco

Sistemi Operativi A.A 2015/2016

Sottosistema I/O

- Uno degli scopi dei Sistemi Operativi è di nascondere all'utente le peculiarità dei dispositivi hardware
- Il sottosistema I/O è responsabile di:
 - Gestione della memoria per I/O per: buffering (memorizzare i dati temporaneamente mentre vengono trasferiti), caching (memorizzare dati in memorie veloci per aumentare la performance), spooling (la sovrapposizione dell'output di un job con gli input di altri job)
 - Interfaccia di uso generale tra device e driver
 - Drivers per dispositivi speciali

Sistemi Operativi A.A 2015/2016

Protezione e Sicurezza

- **Protezione** – ogni meccanismo per controllare l'accesso dei processi/utenti a risorse definite dal Sistema Operativo
- **Sicurezza** – difesa del sistema contro attacchi interni ed esterni
 - Tipo: denial-of-service, worms, viruses, furto di identità, furto di servizi
- I sistemi generalmente distinguono gli utenti per determinare chi può fare cosa
 - Identità degli utenti - User identities (**user IDs**, security IDs) includono nome e un numero associato, uno per ogni utente
 - User ID è associato a tutti i file e processi dell'utente per realizzare il controllo degli accessi
 - Group identifier (**group ID**) permette di identificare un gruppo di utenti e può essere usato per controllare accesso risorse a livello di gruppo, associato a ogni processo/file
 - **Privilege escalation** permette in alcuni casi all'utente di cambiare il proprio id per avere più diritti

Sistemi Operativi A.A. 2015/2016

Ambienti di Elaborazione

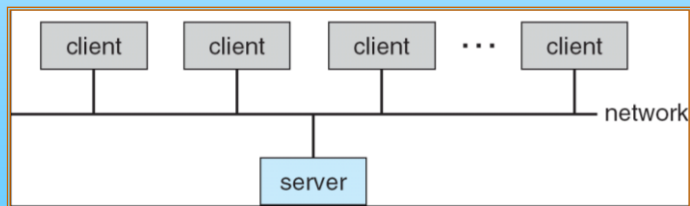
- Personal Computer
 - In ufficio
 - ▶ PCs connessi a una rete locale, terminali connessi a un mainframe o minicomputers forniscono accessi batch e in timesharing
 - ▶ Portali permettono a sistemi in rete e remoti l'accesso alle stesse risorse
 - A casa
 - ▶ Era un singolo sistema
 - ▶ Ora più sistemi in rete dietro firewall

Sistemi Operativi A.A. 2015/2016

Ambienti di Elaborazione (Cont.)

■ Client-Server Computing

- Terminali stupidi soppiantati dai PC
- Molti sistemi sono **servers**, che rispondono a richieste fatte dai **clients**
 - ▶ **Compute-server** fornisce una interfaccia ai client per richiedere servizi (es. database)
 - ▶ **File-server** fornisce una interfaccia ai client per memorizzare e accedere ai file



Sistemi Operativi A.A 2015/2016

Peer-to-Peer Computing

- Un altro modello di sistema distribuito
- P2P non distingue tra client e server
 - Tutti i nodi sono considerati peers (pari)
 - Ogni nodo può agire come client, server o entrambi
 - Un nodo deve connettersi a una rete P2P
 - ▶ Registra il suo servizio su un servizio centrale, o
 - ▶ Fa richieste in broadcast per un servizio e risponde a richieste di servizio usando un *discovery protocol*
 - Esempi sono *Napster*, *Gnutella*, ...

Sistemi Operativi A.A 2015/2016

Struttura dei Sistemi Operativi

Servizi Sistema Operativo

- Un insieme dei servizi offerti dal Sistema Operativo fornisce funzioni utili per l'utente:
 - **Interfaccia Utente** (User interface) – Quasi tutti i sistemi operativi hanno una interfaccia utente (UI)
 - ▶ Varia tra Command-Line (CLI), Graphics User Interface (GUI), Batch
 - **Esecuzione dei programmi** – Il sistema deve poter caricare un programma in memoria ed eseguirlo, terminare la sua esecuzione, sia normale che in caso di errore
 - **Operazioni di I/O** – Un programma in esecuzione può aver bisogno di I/O per l'accesso a file o per interagire con un dispositivo.
 - **Accesso a File-system** – Il File-system è di particolare interesse. I programmi hanno bisogno di leggere/scrivere file e directories, crearle e cancellarle, ricercarle, etc.

Servizi Sistema Operativo (Cont.)

- Un insieme dei servizi offerti dal Sistema Operativo fornisce funzioni utili per l'utente (Continua):
 - **Comunicazioni** – Processi possono scambiare informazioni, su uno stesso calcolatore o tra calcolatori in rete
 - La comunicazione può essere fatta attraverso memoria condivisa o attraverso invio di messaggi (pacchetti inviati dal SO)
 - **Rilevazione errori** – Il SO deve essere sempre attento a possibili errori
 - Possono avvenire nella CPU e memoria, in dispositivi I/O e in programmi utenti
 - Per ogni tipo di errore il SO dovrebbe effettuare l'azione appropriata per garantire una elaborazione corretta e consistente
 - Supporto per il debug può migliorare notevolmente la possibilità di usare efficientemente il sistema da parte dell'utente e del programmatore

Sistemi Operativi A.A. 2015/2016

Servizi Sistema Operativo (Cont.)

- Esiste un altro insieme di servizi del SO che servono a garantire l'uso efficiente del sistema stesso attraverso la condivisione delle risorse
 - **Allocazione risorse** – Quando più utenti o più job sono in esecuzione contemporaneamente, ognuno deve avere a disposizione delle risorse
 - Molti tipi di risorse - Alcuni come cicli di CPU, memoria primaria, e file storage possono avere una allocazione specifica, altri come dispositivi di I/O possono avere una gestione generale di tipo richiesta/rilascio.
 - **Accounting** – Per tenere traccia di quali utenti usano, quanto e quali tipi di risorse di sistema
 - **Protezione e sicurezza** – I possessori delle informazioni memorizzate in un sistema multiutente o in rete possono voler controllare l'uso delle informazioni, processi concorrenti non dovrebbero interferire tra loro
 - **Protezione** assicura che tutti gli accessi al sistema sono controllati
 - **Sicurezza** del sistema dall'esterno richiede l'uso dell'autenticazione degli utenti, si estende alla difesa di dispositivi di I/O esterni dai tentativi di uso non valido.
 - Se un sistema deve essere protetto e sicuro, delle precauzioni devono essere prese in ogni sua parte. Una catena è forte come il suo anello più debole.

Sistemi Operativi A.A. 2015/2016

Interfacce utente del SO - CLI

Command Line Interface permette l'inserimento diretto di comandi

- Alcune volte implementato nel kernel, altre volte implementato nei programmi di sistema
- Alcune volte sono disponibili più tipi di CLI – **shells**
- Fondamentalmente riceve un comando dall'utente e lo esegue
 - ▶ Alcune volte i comandi sono built-in nell'interprete dei comandi, altre volte sono solo nomi dei programmi
 - Nell'ultimo caso aggiungere nuovi comandi non richiede di modificare la shell

Sistemi Operativi A.A 2015/2016

Interfacce Utente del SO - GUI

- Interfaccia user-friendly che usa la metafora della scrivania (**desktop**)
 - Basata su mouse, tastiera, e monitor
 - **Icone** rappresentano file, programmi, azioni, ecc.
 - I pulsanti del mouse premuti su oggetti dell'interfaccia provocano varie azioni (danno informazioni, aprono opzioni, eseguono funzioni, aprono directory/cartelle)
 - Inventato a Xerox PARC (1973)
- Molti sistemi includono interfacce CLI e GUI
 - Microsoft Windows ha GUI con CLI "command" shell
 - Apple Mac OS X ha "Aqua" GUI con sotto un kernel UNIX e sono disponibili le shell unix
 - Solaris ha CLI e delle GUI (Java Desktop, KDE)

Sistemi Operativi A.A 2015/2016

Chiamate di Sistema

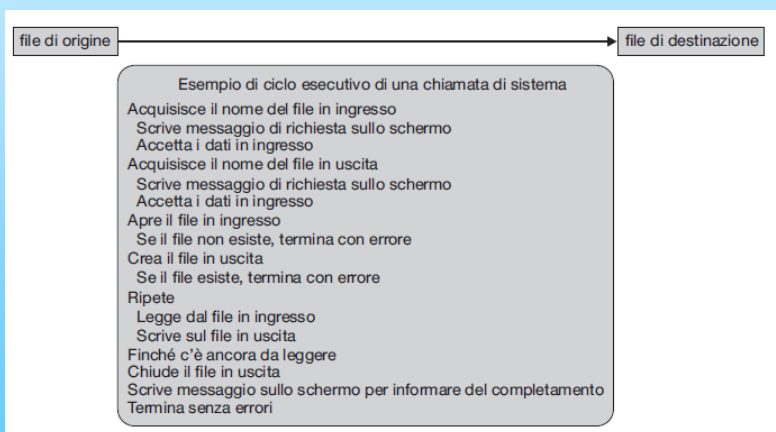
- Interfaccia di programmazione verso i servizi forniti dal SO
- Tipicamente scritte in C/C++
- Principalmente usate dai programmi attraverso **Application Program Interface** (API) di alto livello piuttosto che attraverso l'uso diretto della chiamata di sistema
- Le tre API piu' comuni sono: Win32 API per Windows, POSIX API per sistemi POSIX-based (includendo tutte le versioni di UNIX, Linux, e Mac OS X), e Java API per la Java virtual machine (JVM)

(Nota che I nomi delle chiamate di sistema riportate sono generici)

Sistemi Operativi A.A 2015/2016

Esempio di Chiamata di Sistema

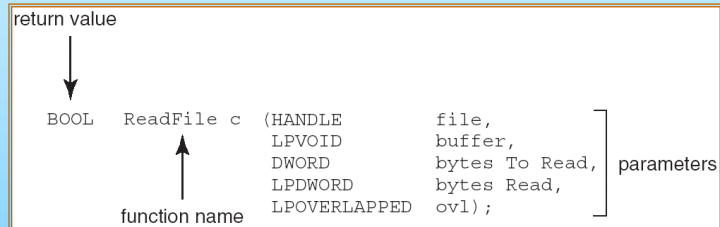
- Sequenza di chiamate di sistema per copiare il contenuto di un file su un altro file



Sistemi Operativi A.A 2015/2016

Esempio di Standard API

- Consideriamo la funzione *ReadFile()*
- Fa parte delle Win32 API— una funzione per leggere da un file



- Parametri:
 - `HANDLE file` — riferisce al file da leggere
 - `LPVOID buffer` — un buffer dove saranno scritti i dati letti dal file
 - `DWORD bytesToRead` — il numero di bytes da leggere da file
 - `LPDWORD bytesRead` — il numero di bytes letti nell'ultima operazione di lettura
 - `LPOVERLAPPED ovl` — indica se deve usare overlapped I/O (asincrono)

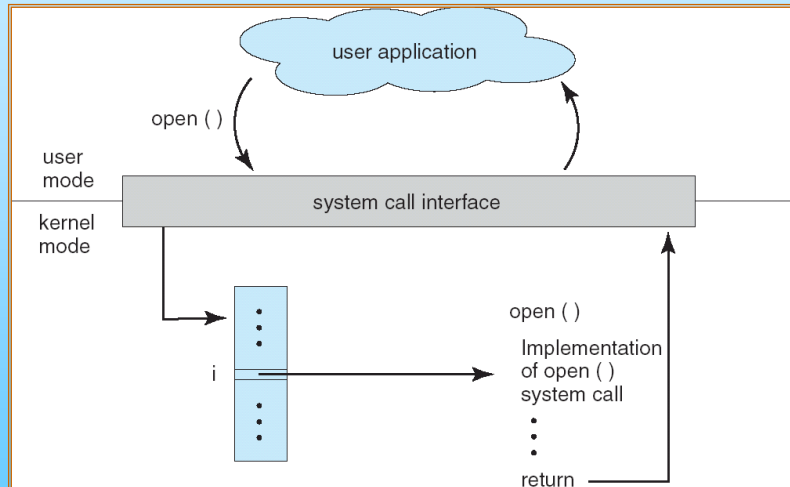
Sistemi Operativi A.A 2015/2016

Implementazione delle Chiamate di Sistema

- Tipicamente un numero e' associato con ogni system call
 - L'interfaccia per le System-call mantiene una tabella indicizzata in base a questi numeri
- L'interfaccia delle chiamate di sistema invoca la chiamata di sistema a livello di kernel e ritorna lo stato della chiamata e ogni valore di ritorno
- Il chiamante non ha necessita' di sapere come la chiamata di sistema e' implementata
 - Deve solo seguire la API e capire cosa il SO fara' come risultato della chiamata
 - I dettagli della interfaccia sono nascosti al programmatore dalle API
 - ▶ Gestite da librerie di supporto a run-time (insieme di funzioni in librerie incluse con il compilatore)

Sistemi Operativi A.A 2015/2016

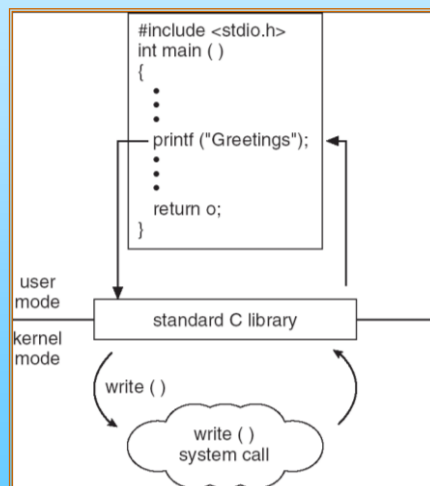
API – Chiamata di Sistema – SO



Sistemi Operativi A.A 2015/2016

Esempio Libreria Standard C

- Programma C che invoca printf() funzione di libreria, che chiama la write() system call



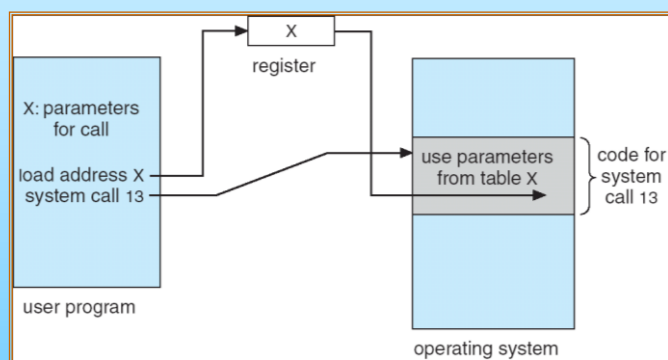
Sistemi Operativi A.A 2015/2016

Passaggio dei Parametri a Chiamata di Sistema

- Tre metodi generali sono usati per passare parametri a SO
 - Più semplice: passare i parametri nei registri della CPU
 - ▶ In alcuni casi si possono avere più parametri che registri
 - Parametri memorizzati in un blocco o tabella in memoria, l'indirizzo del blocco viene passato come parametro in un registro
 - ▶ Usato da Linux e Solaris
 - Parametri *pushed* sullo *stack* dal programma e *popped* dallo stack dal SO
 - Metodi che usano il blocco e lo stack non limitano il numero e la lunghezza dei parametri passati.

Sistemi Operativi A.A 2015/2016

Passaggio parametri con Tabella



Sistemi Operativi A.A 2015/2016

Tipi di Chiamate di Sistema

- Controllo processi
- Gestione File
- Gestione Dispositivi
- Comunicazioni

Sistemi Operativi A.A 2015/2016

Programmi di Sistema

- I programmi di sistema forniscono un ambiente per lo sviluppo di programmi e per la loro esecuzione. Possono essere divisi in programmi per:
 - La manipolazione di file
 - Ottenere informazioni sullo stato
 - La modifica di file
 - Il supporto di linguaggi di programmazione
 - Il caricamento di programmi e loro esecuzione
 - Le comunicazioni
 - Programmi applicativi
- Per la maggior parte degli utenti un sistema operativo è definito dai programmi di sistema disponibili, non dalle chiamate di sistema fornite.

Sistemi Operativi A.A 2015/2016

Programmi di Sistema

- Forniscono un ambiente per lo sviluppo ed esecuzione dei programmi
 - Alcuni sono semplicemente delle interfacce per le chiamate di sistema; altri sono considerevolmente più complessi.
- **Gestione file** – Crea, cancella, copia, rinomina, stampa, lista e generalmente manipola file e directory.
- **Informazione sullo stato**
 - Alcuni chiedono al sistema informazioni – data e ora, quantità di memoria disponibile, uso spazio disco, numero di utenti
 - Altri forniscono informazioni dettagliate per l'analisi della performance, per il logging, e per il debug
 - Tipicamente questi programmi stampano l'output su terminale o altri dispositivi di output
 - Alcuni sistemi implementano un registro – usato per memorizzare e reperire informazioni sulla configurazione

Sistemi Operativi A.A 2015/2016

Programmi di sistema (continua)

- **Modifica di file**
 - Editor di testo per creare e modificare file
 - Comandi speciali per cercare nei file o per effettuare trasformazioni del testo
- **Supporto per i linguaggi di programmazione** – Compilatori, assembleri, debugger e interpreti
- **Caricamento programmi ed esecuzione** – Absolute loaders, relocatable loaders, linkage editors, e overlay-loaders, programmi per il debug per linguaggi di alto livello e per linguaggio macchina
- **Comunicazione** – Fornisce meccanismi per creare connessioni virtuali tra processi, utenti e calcolatori
 - Permettere agli utenti di scambiarsi messaggi, navigare pagine web, inviare e-mail, collegarsi in modo remoto, trasferire file tra macchine diverse.

Sistemi Operativi A.A 2015/2016

Progettazione e implementazione di Sistemi Operativi

- Progettazione e implementazione di un sistema operativo è un problema difficile, ma alcuni approcci si sono dimostrati di successo
- La struttura interna di sistemi operativi differenti può variare moltissimo
- Si inizia definendo gli obiettivi (goals) e le caratteristiche del sistema
- Dipende dall'hardware scelto e dal tipo di sistema (a lotti, in time-sharing, mono/multiutente, real time o uso generale)
- *User goals* e *System goals*
 - User goals – il sistema operativo dovrebbe essere facile da usare, facile da imparare, affidabile, sicuro e veloce
 - System goals – il sistema operativo dovrebbe essere facile da progettare, implementare e mantenere, così come essere flessibile, affidabile e efficiente

Sistemi Operativi A.A 2015/2016

Progettazione e implementazione di Sistemi Operativi (continua)

- Un principio importante è di separare
 - Criteri o politiche:** Cosa sarà fatto?
 - Meccanismi:** Come sarà fatto?
- I meccanismi determinano come fare qualcosa, i criteri decidono cosa sarà fatto
 - La separazione tra criteri e meccanismi è un principio molto importante permette la massima flessibilità se i criteri cambiano nel tempo
- Esempio:
 - Il *Timer* è un **meccanismo**
 - La decisione di quanto tempo assegnare ad un processo riguarda un **criterio**
- Per l'assegnazione delle risorse sono importanti i criteri

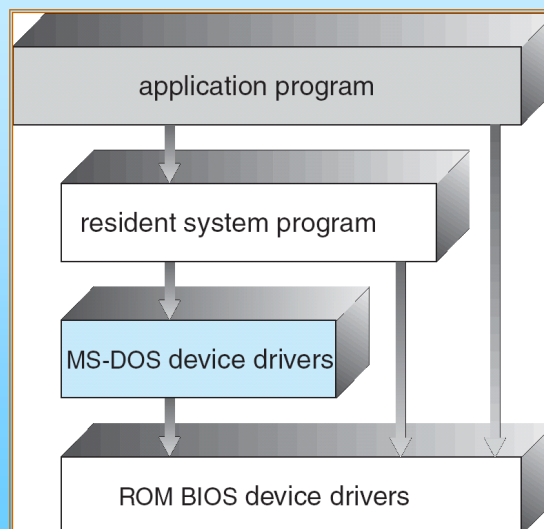
Sistemi Operativi A.A 2015/2016

Struttura semplice

- MS-DOS – scritto per fornire il maggior numero di funzionalità nel minor spazio
 - Non diviso in moduli
 - Sebbene MS-DOS abbia una struttura, le sue interfacce e funzionalità non sono ben separate

Sistemi Operativi A.A 2015/2016

MS-DOS Struttura a strati



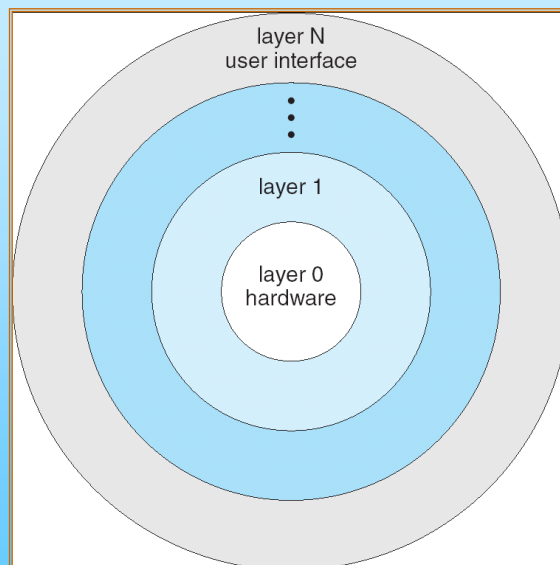
Sistemi Operativi A.A 2015/2016

Approccio stratificato

- Il sistema operativo è diviso in un numero di strati (layer), ognuno costruito sulla base degli strati inferiori. Lo stato più basso (layer 0), è l'hardware; lo stato più alto (layer N) è la interfaccia utente.
- Con la modularità, gli strati sono selezionati in modo che ognuno usi funzioni e servizi dei soli strati di livello inferiore
- Facile da verificare e testare
- Difficile definire bene gli strati
- Può introdurre inefficienze dovute all'attraversamento dei vari strati in una chiamata.

Sistemi Operativi A.A 2015/2016

Sistema Operativo Stratificato



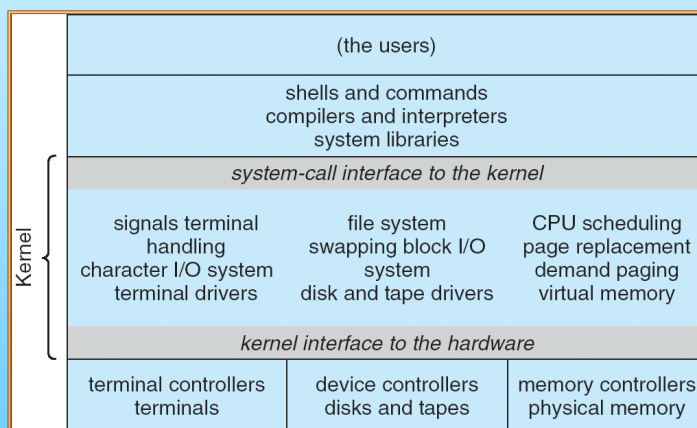
Sistemi Operativi A.A 2015/2016

UNIX

- UNIX – limitato dalle funzionalità del hardware usato, il sistema operativo UNIX originale aveva una strutturazione limitata. UNIX consiste di due parti separate:
 - Programmi di sistema
 - Il kernel
 - ▶ Consiste della parte sotto l'interfaccia delle chiamate di sistema e sopra l'hardware
 - ▶ Fornisce il file system, scheduling della CPU, gestione della memoria, ed altre funzioni del sistema operativo; un grande numero di funzioni su un livello

Sistemi Operativi A.A 2015/2016

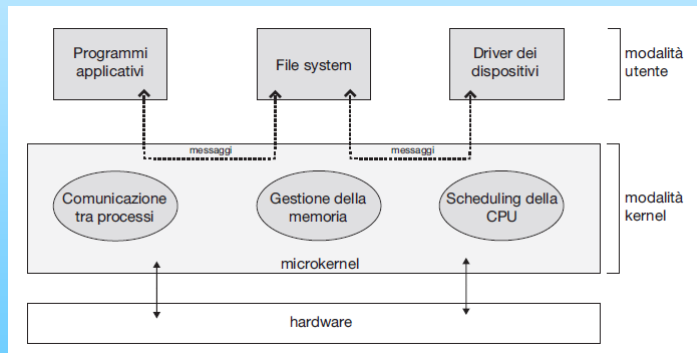
Struttura di UNIX



Sistemi Operativi A.A 2015/2016

Struttura a Microkernel

- Sposta il più possibile dal kernel nello spazio “utente”
- La comunicazione tra i moduli utente usa l'invio di messaggi



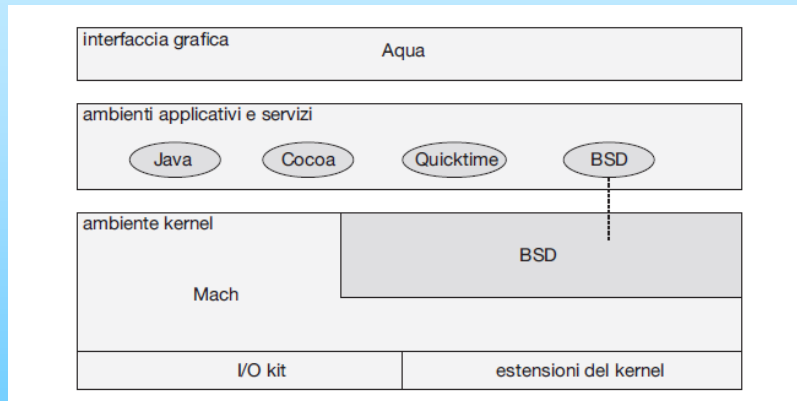
Sistemi Operativi A.A 2015/2016

Struttura a Microkernel

- Benefici:
 - Più facile da estendere
 - Più facile portare il sistema operativo su nuove architetture hardware
 - Più affidabile (meno codice in esecuzione in modo kernel)
 - Più sicuro
- Demeriti:
 - Diminuzione di performance dovuto alla comunicazione tra spazio utente e kernel

Sistemi Operativi A.A 2015/2016

Mac OS X Structure



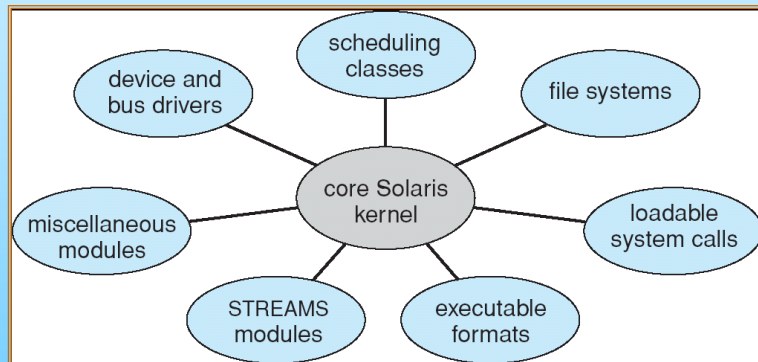
Sistemi Operativi A.A 2015/2016

Moduli

- La maggior parte dei moderni sistemi operativi implementano i moduli kernel
 - Usa un approccio object oriented
 - Ogni componente base è separato
 - Ogni modulo comunica con gli altri moduli attraverso interfacce ben definite
 - Ogni modulo viene caricato in base alle necessità nel kernel
- Approccio simile alla stratificazione ma più flessibile

Sistemi Operativi A.A 2015/2016

Approccio Modulare di Solaris



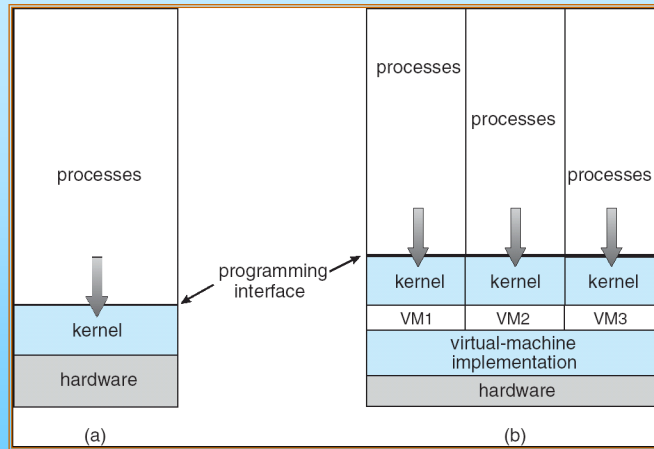
Sistemi Operativi A.A 2015/2016

Macchine Virtuali

- Una macchina virtuale porta a naturale conclusione l'approccio stratificato. Tratta l'hardware e il sistema operativo come se fosse hardware
- Una macchina virtuale fornisce una interfaccia identica all'hardware sottostante
- Il sistema operativo crea l'illusione di disporre di più macchine ognuna in esecuzione sul suo processore e con la propria memoria (virtuale)
- Le risorse del computer fisico sono condivise per creare le macchine virtuali
 - Lo scheduling della CPU crea l'illusione che ogni utente abbia il proprio processore
 - Il file system può fornire uno spazio di memorizzazione per ogni macchina virtuale
- Nate per mainframe IBM nel 1972 (IBM VM/370), oggi portate a nuovo splendore (VMware)
- Oggi utilizzate nei datacenter per utilizzare al meglio le risorse di calcolo

Sistemi Operativi A.A 2015/2016

Macchine Virtuali (Continua)



(a) macchina non-virtuale (b) macchina virtuale

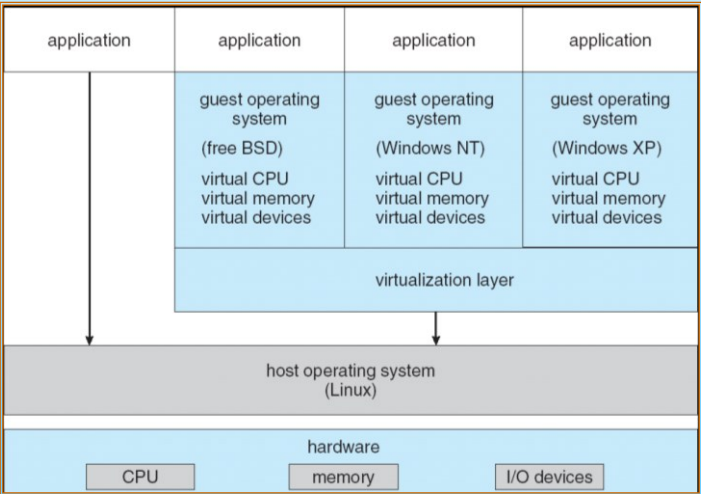
Sistemi Operativi A.A 2015/2016

Macchine Virtuali (Continua)

- Le macchine virtuali forniscono la completa protezione delle risorse del sistema, infatti ogni macchina virtuale è isolata dalle altre macchine virtuali. L'isolamento, comunque, non permette la condivisione diretta di risorse.
- Una macchina virtuale è il modo perfetto per fare ricerca e sviluppo sui sistemi operativi. Lo sviluppo viene fatto su una macchina virtuale invece che sulla macchina fisica in modo da non disturbare il normale uso del sistema.
- La virtualizzazione di una macchina è difficile da implementare dovuto allo sforzo necessario a fornire un duplicato esatto della macchina fisica sottostante e a fare questo in modo efficiente
- Problema: eseguire codice kernel (del SO ospite) in modo utente (nel SO ospitante)

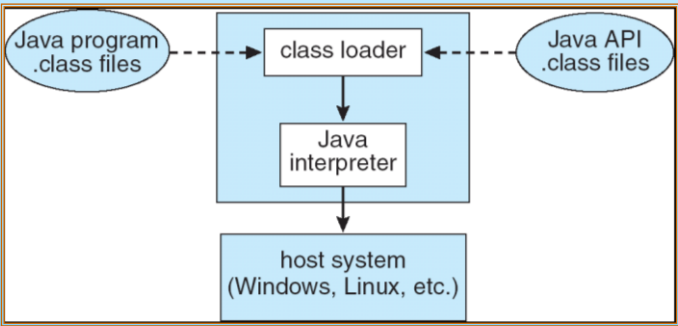
Sistemi Operativi A.A 2015/2016

Architettura VMware



Sistemi Operativi A.A 2015/2016

Macchina Virtuale Java



Sistemi Operativi A.A 2015/2016