SISTEMI

I Sistemi Operativi

Prof. Viglietti Francesco

Classe 4 B info

A.S. 2010-11

Intro ai Sistemi Operativi

Software

Programmi di sistema

Controllano il comportamento del pc

Programmi applicativi

Risolvono i problemi degli utenti

Sistema Operativo (OS)

Strato di software al di sopra dell'hardware che controlla tutte le parti del sistema e presenta all'utente una interfaccia (macchina astratta) più semplice da comprendere e programmare.

Controlla tutte le risorse del computer:

- memoria
- CPU
- dischi
- periferiche

I primi OS risiedevano su ROM (Commodore) o si caricavano da floppy (MS-DOS). Oggi devono essere installati su disco fisso. Durante l'installazione viene richiesta una configurazione in base all'HW della macchina o alle esigenze dell'utente.

La shell è l'interfaccia di interazione con l'utente, può essere a riga di comando (Prompt) o grafica (GUI)

Obiettivi di un OS

Efficiente: Se il OS raggiunge un alto **throughput** (mole di lavoro che la CPU puo completare in un certo tempo), ed un basso tempo medio di **turnaround** (Tempo di terminazione di un processo)

Robusto: reagisce in modo graduale ad eventuali malfunzionamenti che si verificano in applicazioni isolate, senza compromettere la funzionalità dell'intero sistema.

Scalabile: Se è in grado di usare risorse che vengono aggiunte al sistema. Importante nei sistemi multiprocessore.

Espandibile: Se è in grado di adattarsi alle nuove tecnologie e presenta la capacità di svolgere compiti che vanno oltre quelli pensati inizialmente.

Portabile: quando è progettato per funzionare su diverse configurazioni HW

Sicuro: se impedisce agli utenti ed al SW di accedere a sevizi e risorse senza le opportune autorizzazioni. Importante il concetto di **protezione** che si riferisce ai meccanismi che realizzano la politica di sicurezza del sistema.

Interattivo: Se permette alle applicazioni di rispondere velocemente alle azioni dell'utente o agli eventi.

Usabile: Se è in grado di essere adatto ad un gran numero di utenti

I tipi di OS

I OS possono essere classificati in base al numero di utenti e di processi che il sistema puo gestire contemporaneamente. Abbiamo:
single user, single task: un solo utente, un solo processo (es: DOS)
single user, multitasking: un solo utente, ma piu applicazioni che possono funzionare “contemporaneamente” (fino a Win XP, Mac OS)
multiuser: piu utenti possono utilizzare lo stesso computer (Linux, Win Vista/7) al computer sono collegati più terminali.
Real Time OS: sono sistemi usati per il controllo di processi di tipo industriale in cui la cosa piu importante e il tempo di risposta.
Network OS: Sono sistemi single user, che permettono ad altri utenti dello stesso sistema, di collegarsi attraverso la rete per utilizzarne alcune risorse.Tutti gli utenti si possono collegare al sistema attraverso propri pc ognuno con un proprio OS.
Alcuni OS possono simulare HW o OS diversi da quello presente, ogni utente può usare un computer virtuale (es. DOS su Windows)

I componenti principali OS

I componenti principali del OS sono contenuti nel *Kernel* e sono:
*Scheduler Processi (**Sch.P.**)*: determina quando e per quanto tempo ogni singolo processo sarà in esecuzione su un certo processore
*Gestore della memoria Principale (**G.M.**)*: determina quando e quanta memoria è allocata ai processi e gestisce i meccanismi relativi all'esaurimento della memoria principale (Memoria Virtuale)
*Gestore dispositivi di I/O (**I/O**)*: soddisfa le richieste da e verso i dispositivi HW (periferiche)
*Gestore della comunicazione fra processi (**IPC**):* permette la comunicazione fra i processi
*Gestore del File System (**F.S.**):* organizza raccolte di dati su dispositivi di memorizzazione e fornisce un interfaccia per accedere a tali dati

I servizi del OS

I servizi principali per l'utente sono la gestione dei file e l'esecuzione dei programmi.
Identificazione dei file (nome ed estensione)
Directory o cartelle (contenitori di file e cartelle, gerarchia ad albero, la directory radice viene rappresentata da: \ in Win e / in Linux)
Pathname (percorso relativo od assoluto)

L'esecuzione di un programma viene chiamata *processo* o *task*;
il *programma* è la descrizione delle istruzioni da eseguire.
il *processo* è la sequenza di azioni compiute durante l'esecuzione.

Meccanismi e sistemi di protezione

La CPU può operare in 2 modalità differenti:

kernel: quando esegue le routine del sistema, può accedere a qualunque area della memoria.

Utente: se esegue il programma dell'utente, può accedere all'area di memoria riservata all'utente.

Nei sistemi multiutenti bisogna creare gli account e definire i privilegi di ciascuno.

L'**administrator** gestisce la macchina e gli altri utenti.

Per il controllo il sistema offre meccanismi di auditing che prevedono file di log per le operazioni degli utenti.

Protezione di file e cartelle.

Ogni file ha un proprietario che ne definisce i permessi.

La concorrenza

Per esecuzione concorrente si intende l'esecuzione di due o più azioni parallele. Se il sistema è multiprocessore allora le due azioni possono eseguirsi contemporaneamente (parallelismo reale), altrimenti le due azioni avanzeranno in modo alternato nel tempo (interleaving o quasi parallelismo).

I processi eseguiti in parallelo hanno interazioni tra loro.

Due o più processi *interferiscono* tra loro se si condizionano per l'uso delle risorse.

Due o più processi *cooperano* tra loro se comunicano per ottenere uno scopo comune.

Per i processi cooperanti, sono richiesti i seguenti meccanismi di interazione tra processi:

- condivisione dei dati: permette lo scambio di informazioni tra dati comuni, l'accesso è disciplinato dalla mutua esclusione.
- sincronizzazione: scambio di informazioni temporali in modo da stabilire la precedenza tra le azioni svolte da processi diversi.
- comunicazione: scambio di informazioni con messaggi

Riassumendo: i compiti di un OS

- accettare le richieste degli utenti (Shell)
- eseguire i processi, gestendone le interazioni ed assegnandone le risorse necessarie (Kernel)
- permettere lo sviluppo dei programmi (tool e API)

I tool di utilità sono editor, compilatori,... tutti atti allo sviluppo dei programmi

Le API sono le primitive per lo sviluppo di applicazioni.

Architettura dei OS 1

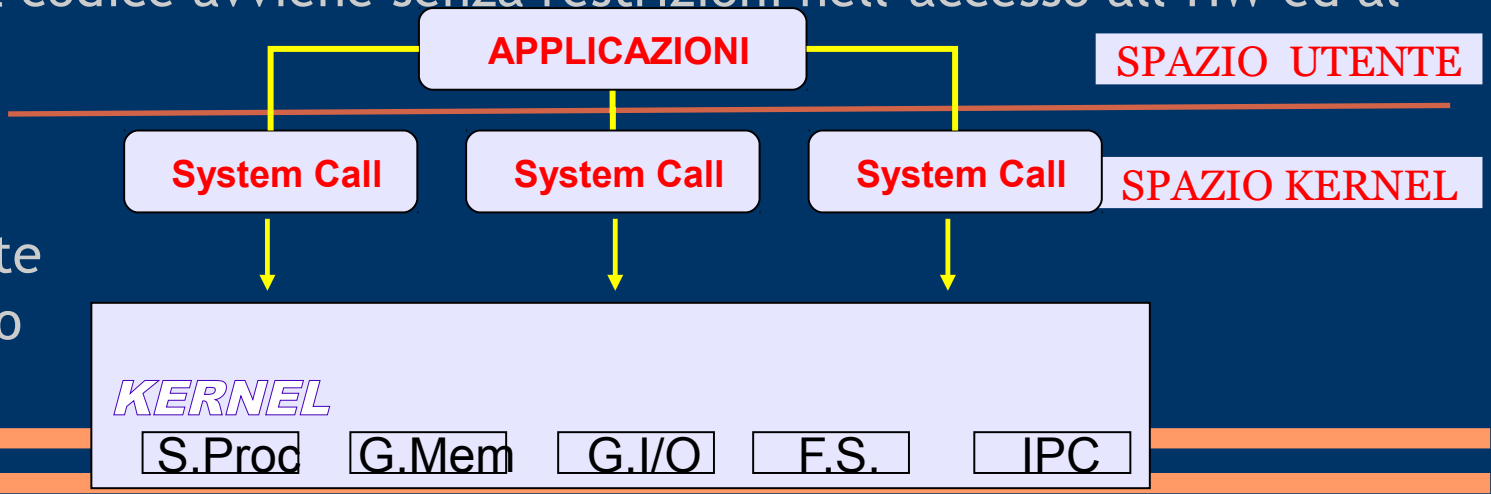
I OS oggi sono complessi, in quanto forniscono una vasta gamma di servizi e devono gestire una gran quantità di risorse HW e SW. Le architetture aiutano a gestire tali complessità fornendo un'organizzazione per i componenti del sistema e specificando i privilegi con cui eseguirli.

Un OS può essere progettato strutturandolo in diversi modi:
Struttura Monolitica, Struttura Stratificata, Microkernel.

Nella struttura monolitica Ogni componente del OS (Scheduler, Gestore memoria, Gestore I/O, IPC e File System) fa parte del Kernel

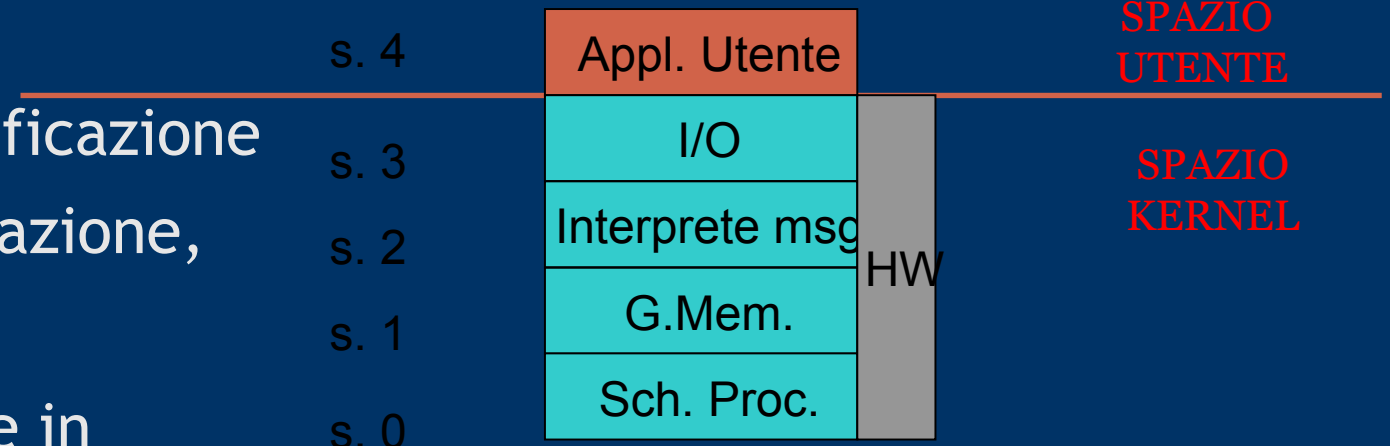
PRO: Efficienza, grazie alla comunicazione diretta fra i componenti principali del OS

CONTRO: Difficoltà ad isolare fonti di errore e malfunzionamenti. Sistema poco robusto. L'esecuzione del codice avviene senza restrizioni nell'accesso all'HW ed al SW del sistema, per cui sono particolarmente esposti a danni derivanti da codice accidentalmente o intenzionalmente errato



Architettura dei OS 2

Struttura stratificata → si cerca di raggruppare in strati i componenti che svolgono funzioni simili. La comunicazione può avvenire solo fra starti adiacenti. Gli strati inferiori forniscono servizi a quelli superiori per mezzo di un interfaccia che maschera il modo in cui il lavoro viene svolto.



PRO: Modularità. Semplificazione delle operazioni di validazione, debug e modifica

CONTRO: Poco efficiente in termini di velocità di esecuzione di un servizio. L'esecuzione del codice avviene senza restrizioni nell'accesso all'HW ed al SW del sistema, per cui sono particolarmente esposti a danni derivanti da codice accidentalmente o intenzionalmente (virus) errato

Architettura dei OS 3

Struttura a microkernel → Fornisce solo un piccolo numero di servizi nel tentativo di mantenere il kernel piccolo e scalabile

PRO: Elevata Modularità che lo rende espandibile scalabile e portabile. Robustezza per il fatto che non essendo inclusi nel kernel tutti i componenti principali, il blocco di uno non compromette la funzionalità dell'intero sistema

CONTRO: Elevato numero di comunicazioni tra i moduli che finisce per peggiorare le prestazioni del sistema