

CAP. 3 - STRUTTURE DEI SISTEMI OPERATIVI

Sistemi Operativi

1

Vittorio Cortellesa, 2002-2003

Componenti comuni di un OS

1. Gestione dei processi
2. Gestione della memoria centrale
3. Gestione della memoria secondaria
4. Manipolazione dei file
5. Gestione dell'I/O
6. Sistema di protezione
7. Gestione di rete
8. Interprete dei comandi

Sistemi Operativi

2

Vittorio Cortellesa, 2002-2003

1. Gestione dei processi

Un *processo* e' un programma in esecuzione

Un *processo utente* deriva da un programma eseguito da un utente; un *processo di sistema* corrisponde ad una routine di OS

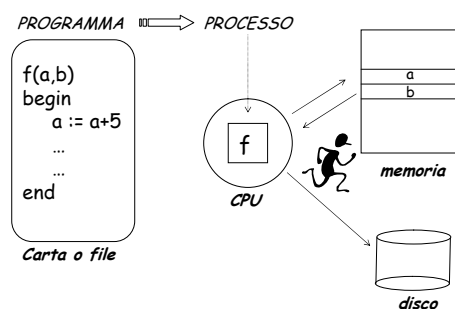
Un processo ha bisogno di *risorse*, quali tempo di CPU, memoria, files e dispositivi di I/O per svolgere il suo compito

Sistemi Operativi

3

Vittorio Cortellesa, 2002-2003

Programma e processo



Sistemi Operativi

4

Vittorio Cortellesa, 2002-2003

Compiti di OS rispetto ai processi

Creazione e cancellazione

Sospensione e ripristino (scheduling)

Sincronizzazione (deadlock)

Comunicazione

Sistemi Operativi

5

Vittorio Cortellesa, 2002-2003

2. Gestione della memoria centrale

Memoria : un array di parole o bytes, ognuna con il suo indirizzo

- Velocemente accessibile
- Di piccola taglia
- Volatile
- Condivisa tra la CPU e i dispositivi di I/O

Sistemi Operativi

6

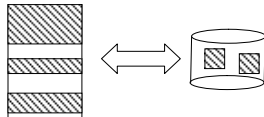
Vittorio Cortellesa, 2002-2003

Compiti di OS rispetto alla memoria

Tenere traccia di quali parti della memoria sono correntemente usate e da chi

Decidere quali processi caricare da disco quando diventa disponibile spazio di memoria

Allocare e deallocare spazio di memoria quando necessario



Sistemi Operativi

7

Vittorio Cortellessa, 2002-2003

3. Gestione della memoria secondaria

La **memoria secondaria** serve da **backup per la memoria centrale** (troppo piccola e volatile per conservare informazioni in maniera permanente)



In genere si tratta di **dischi**, aventi caratteristiche proprie quali la velocità di accesso, la capienza, etc...

Sistemi Operativi

8

Vittorio Cortellessa, 2002-2003

Compiti di OS rispetto alla memoria secondaria

Gestione dello spazio libero

Allocazione della memoria

Scheduling delle richieste

Sistemi Operativi

9

Vittorio Cortellessa, 2002-2003

4. Manipolazione dei file

Un **file** è una **collezione di informazioni** (in qualche modo **in relazione** tra esse) definita dal suo creatore

Un file comunemente contiene o **programmi** o **dati** o **altri file** (directory)

Il file è lo **strumento di astrazione** dalle memorie secondarie

Sistemi Operativi

10

Vittorio Cortellessa, 2002-2003

Compiti di OS rispetto ai file

Creazione e cancellazione di file/directory

Primitive per la manipolazione (open, read, etc.)

Mapping di file a dispositivi di memoria

Backup di files su dispositivi appositi

Sistemi Operativi

11

Vittorio Cortellessa, 2002-2003

5. Gestione dell'I/O

- Sistema di buffering, caching e spooling
 - **Buffering**: preparare i dati di I/O in predisposti registri e aree di memoria
 - **Caching**: conservare in memorie veloci i dati usati più di frequente per I/O (ex. indirizzi per I/O mapped)
 - **Spooling**: virtualizzazione dei dispositivi di I/O mediante uso di aree di memoria per caricare e scaricare informazioni
- **Driver** per ogni tipo di dispositivo: routine di I/O che colloquia con il controller del dispositivo
- **Interfaccia generale per i driver**: programma che si invoca per iniziare qualsiasi operazione di I/O

Sistemi Operativi

12

Vittorio Cortellessa, 2002-2003

6. Sistema di protezione

Una *risorsa* e' un'entita' *passiva*, quindi non e' in grado di proteggersi da sola contro usi impropri ed errori

Piu' in generale un *sistema di protezione* si occupa di *controllare gli accessi* di programmi, processi, utenti a OS e risorse



Sistemi Operativi

13

Vittorio Cortellessa, 2002-2003

Compiti di OS per la protezione

Distinguere tra *usi/utenti autorizzati* e non

Specificare i *controlli* da imporre

Fornire un meccanismo per il *rispetto dei vincoli* imposti dalle protezioni

Sistemi Operativi

14

Vittorio Cortellessa, 2002-2003

7. Gestione di rete

Un *sistema distribuito* e' una collezione di "elaboratori" (possibilmente *non omogenei* tra loro) che *non condividono memoria e clock*, e che sono collegati mediante una *rete di comunicazione*

Sistemi Operativi

15

Vittorio Cortellessa, 2002-2003

Compiti di OS per una rete

Fornire accesso alle risorse di rete da parte degli utenti



Velocizzazione dell'esecuzione

Maggiore *disponibilita'* di informazioni

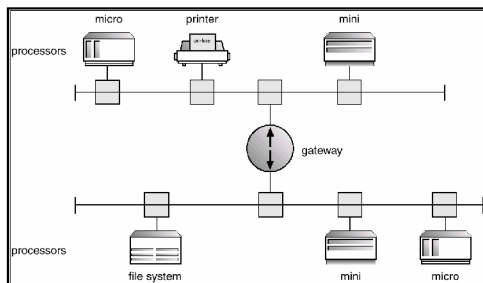
Maggiore *affidabilita'*

Sistemi Operativi

16

Vittorio Cortellessa, 2002-2003

Un esempio di Local Area Network



Sistemi Operativi

17

Vittorio Cortellessa, 2002-2003

8. Interprete dei comandi (shell)

La *shell* e' il *programma* che si occupa di accettare, interpretare ed *eseguire il prossimo comando*



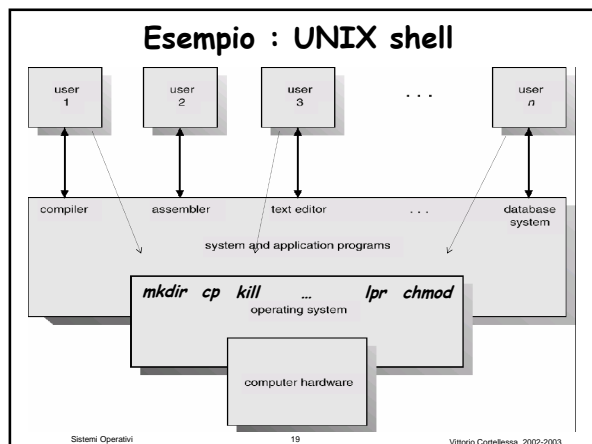
L'insieme di comandi costituisce il *linguaggio di colloquio* tra *utente* e OS

Piu' o meno *facili e user-friendly* shell, a seconda di OS

Sistemi Operativi

18

Vittorio Cortellessa, 2002-2003



Categorie di comandi

TUTTE LE PRECEDENTI DA 1. A 7.!!!

- Gestione dei processi
- Gestione della memoria centrale
- Gestione della memoria secondaria
- Manipolazione dei file
- Gestione dell'I/O
- Sistema di protezione
- Gestione di rete

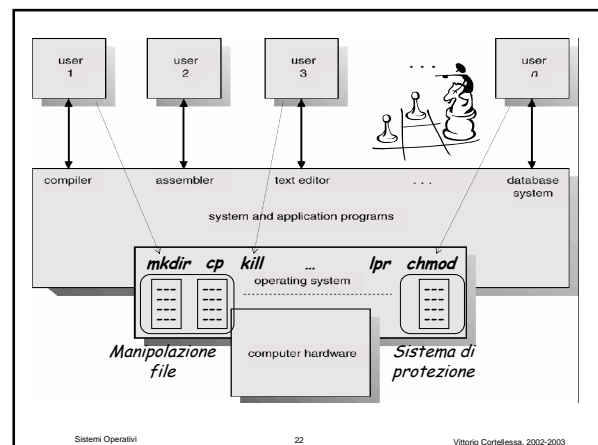
Sistemi Operativi 20 Vittorio Cortellessa, 2002-2003

Modalita' di interpretazione comandi

Interna : l'interprete nel OS salta ad una locazione *interna al suo codice* che contiene il codice corrispondente al comando richiesto (*fissa ma sicura*)

Esterna : il nome del comando rappresenta il nome di *un file che contiene il codice* corrispondente al comando richiesto (*espandibile ma rischiosa*)

Sistemi Operativi 21 Vittorio Cortellessa, 2002-2003



Riassumendo quindi...

Servizi comuni di un OS : *convenienza*

- Esecuzione di programmi
- Operazioni di I/O
- Gestione del File System
- Comunicazioni
- Rilevamento di errori

Funzioni comuni di un OS : *efficienza*

- Allocazione delle risorse
- Contabilizzazione dell'uso delle risorse
- Protezione

Sistemi Operativi 23 Vittorio Cortellessa, 2002-2003

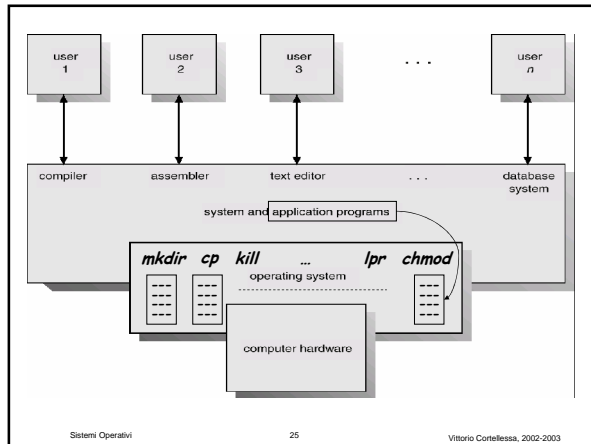
System Calls

Forniscono l'*interfaccia* tra un programma in esecuzione e OS

La maggior parte dei moderni linguaggi di programmazione permette la *chiamata diretta di system call* nel linguaggio stesso

Ad ogni system call *corrisponde* in genere un *comando di OS*, che quindi e' disponibile anche direttamente all'utente in *shell*

Sistemi Operativi 24 Vittorio Cortellessa, 2002-2003



Esempio di programma applicativo con uso di system call : lettura del contenuto di un file e scrittura in un altro file

Sistemi Operativi 26 Vittorio Cortellessa, 2002-2003