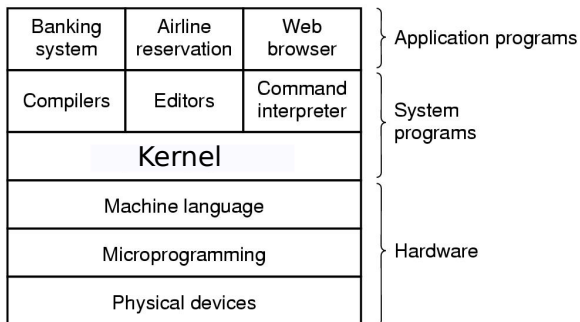


# Sistemi Operativi: Concetti generali

# Concetti generali



- complessità di un SO: interazioni
- regola fondamentale: un SO deve essere utilizzato per uno scopo preciso

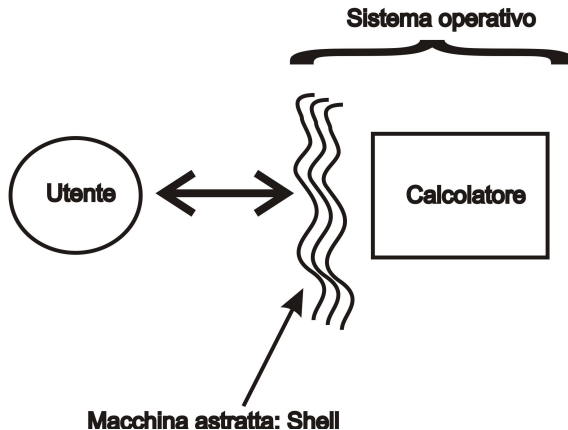
# Servizi fondamentali di un Sistema Operativo

- **Creazione, esecuzione e terminazione dei programmi utente.**
- **Operazioni di Ingresso Uscita.**
- **Manipolazione di file.**
- **Gestione degli errori di tipo hardware, di I/O, generati dall'utente.**
- **Allocazione delle risorse: sono elementi hardware come CPU, memoria file, canali di I/O o elementi software come strutture dati o codice rientrante.**
- **Contabilità e statistiche sull'uso delle risorse: costo e prestazioni.**
- **Protezione e sicurezza.**

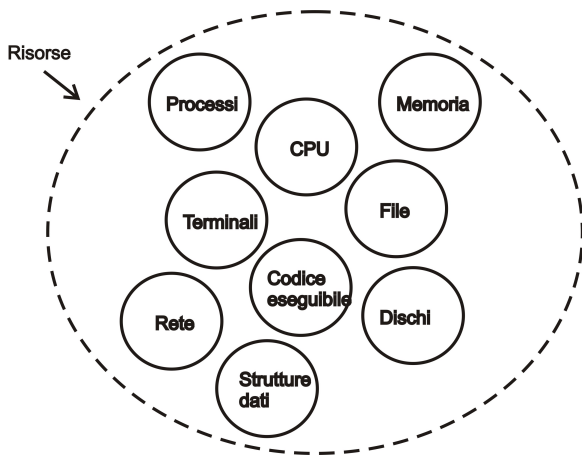
# Classificazione dei Sistemi Operativi secondo l'utilizzazione

- **Utilizzo di tipo Generale (General Purpose).**
- **Utilizzo di tipo Batch (Senza intervento manuale).**
- **Utilizzo di tipo Interattivo (Time sharing).**
- **Utilizzo dedicato (Sistemi Operativi in Tempo Reale).**
- **Sistemi Transazionali (Data base).**
- **Sistemi Operativi Distribuiti (Grid, Cluster..).**

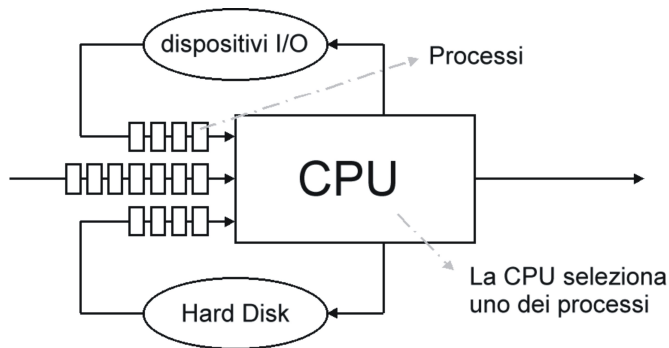
# Sistema operativo come macchina astratta



# Sistema operativo come gestore di risorse



# Sistema operativo come architettura



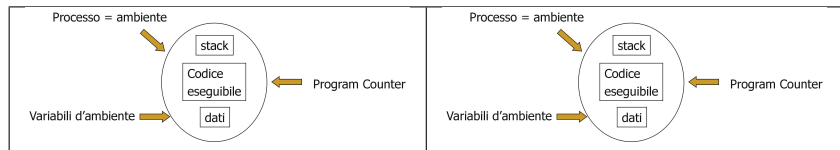
# Processi

- **Programma/Processo.**
- **Processo come ambiente. Codice eseguibile (rientrante, non rientrante) + stack + dati (variabili d'ambiente).**
- **Concorrenza.**
- **Schedulazione.**
- **Cambiamento di contesto (context switch).**
- **Comunicazione tra processi (Inter Process Communication, IPC).**
- **Strutture dati dei processi (Process Control Block, Process Table).**

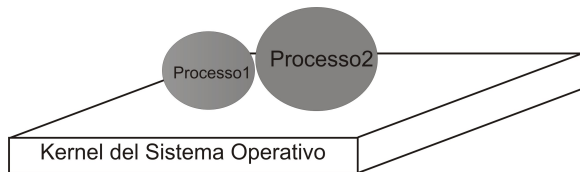


# Processi

## Due processi concorrenti



**ATTENZIONE: i processi NON condividono lo spazio interno!**



# Descrittori di processi

Descrittore di processo: struttura Process Control Block (PCB)

Stato	Registri	Allocazione Memoria	Scheduling	Identificatori	I/O	File aperti	Statistiche uso Risorse
-------	----------	---------------------	------------	----------------	-----	-------------	-------------------------

Lista di descrittori



# Concorrenza di processi

## Qualche vantaggio della concorrenza

- cattura la simultaneità insita nel sistema (esempio: accessi simultanei ad un Web server)
- gestisce contemporaneamente dispositivi hardware diversi con diversi tempi di risposta
- migliora le prestazioni temporali

## Strumenti di basso livello per introdurre la concorrenza

**Coroutine** due routine eseguono in concorrenza trasferendosi il controllo della esecuzione a vicenda mediante primitive linguistiche chiamate 'Transfer' o 'Resume'.

Molti linguaggi offrono Coroutine native ( Modula-2, Oberon, Perl 6, Prolog, Python, Ruby). Altrimenti ( C, C++, Java, ...) → classi e librerie esterne.

Attenzione: ogni coroutine ha il suo stack separato

# Concorrenza tra processi


Strumenti di basso livello per introdurre la concorrenza:

**Concetto della coroutine** Una applicazione rilascia volontariamente l'esecuzione per consentire l'esecuzione di un'altra applicazione.

**Cooperative multitasking** I programmi rilasciano esplicitamente il controllo al Sistema Operativo in occorrenza di determinati eventi. Il Sistema Operativo decide quale programma eseguire.

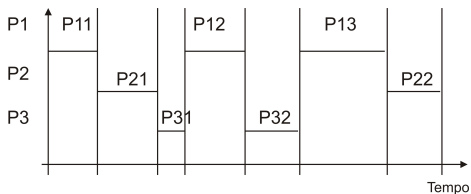
Usata da Microsoft Windows prima di Windows NT e da Mac OS prima di Mac OS X. Schedulazione sincrona.

**Multiprogrammazione** . Il context switch scatta quando una applicazione fa una richiesta di I/O (es. disco) e quando la richiesta viene risolta (tramite interrupt)

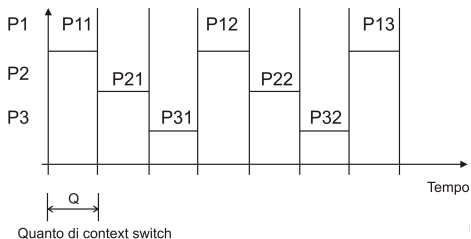
**Time Sharing** Il Sistema Operativo interrompe i processi in istanti che i processi non conoscono. Schedulazione asincrona. 

# Schedulazione dei processi

## Schedulazione generica

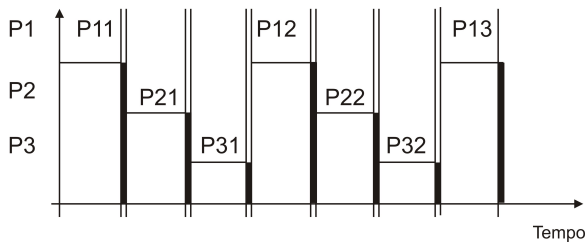


## Schedulazione a divisione di tempo (time sharing)



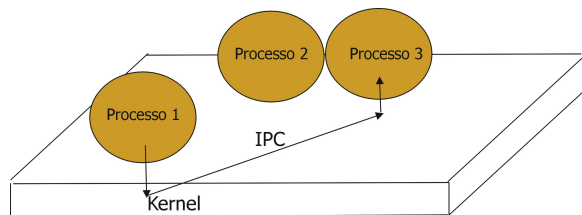
# Schedulazione dei processi

## Contest switch nella schedulazione time sharing



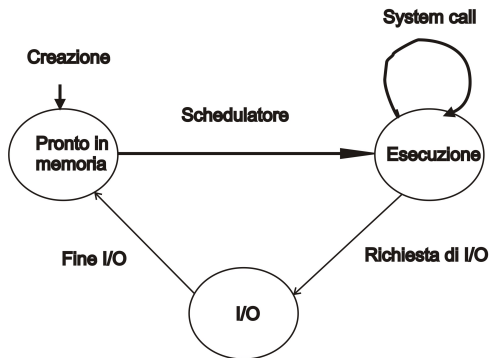
# Comunicazione tra processi concorrenti

Primitive per 'attraversare' i confini processo/kernel



# Stati di un processo

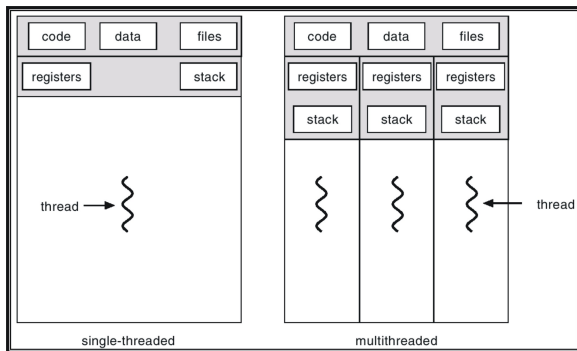
## Sistema monoprocesso/multiprocesso





# Concorrenza a Thread

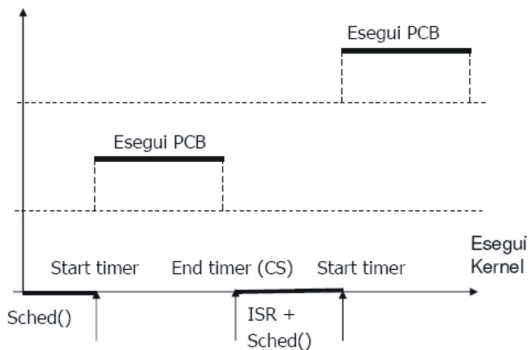
Thread: simili ai processi, MA condividono lo spazio interno!



# Schema della implementazione del time sharing

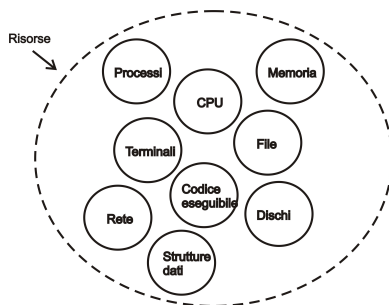
```
Schedulatore()  
{  
    while(true)  
        GestisciCodaProcessi();  
}  
GestisciCodaProcessi()  
{  
    Seleziona_ed_EstraiPCB();  
    StartTimer();  
    EseguiPCB();  
}  
GestoreInterruptTimer()  
{  
    SalvaContesto(); //in PCB  
    Schedulatore();  
}
```

# Schema della implementazione del time sharing



ISR=Interrupt Service Routine

# Risorse



- ✓ **Condivisibili**
- ✓ **Non condivisibili**
- ✓ **Riutilizzabili**
- ✓ **Non riutilizzabili**
- ✓ **Sottraibili**

# Risorse

**Strozzature:** situazione in cui una risorsa viene utilizzata molto pi delle altre ; il sistema operativo pu essere bloccato sull'utilizzo di una determinata risorsa.

**Anelli di reazione:** situazione in cui l'utilizzo errato di una risorsa provoca per reazione un errato utilizzo di altre risorse.

**Unitá centrale:** fornita dalla schedulazione che ritorna periodicamente il controllo al nucleo

**Memoria:** protezione da memoria scrivibile/leggibile e controllo di indirizzo