

## Approfondimento

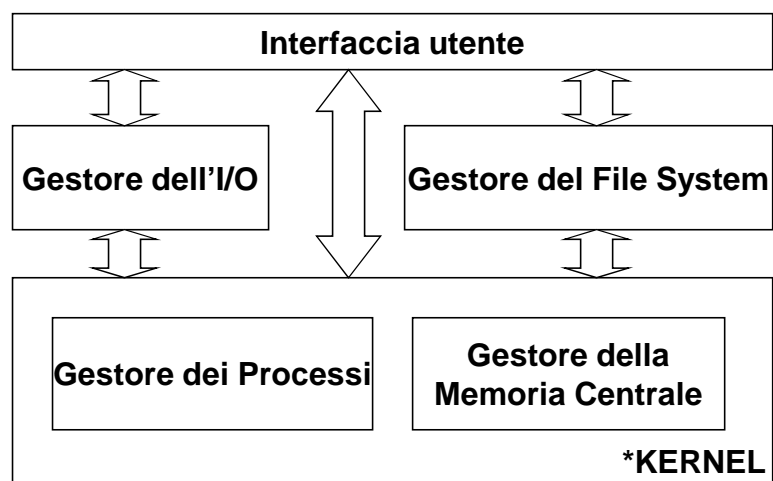
# Sistemi Operativi Kernel

## Kernel del Sistema Operativo

- **Kernel (*nocciolo, nucleo*)**  
Contiene i programmi per la gestione delle funzioni base del calcolatore
- **Kernel suddiviso in moduli. Ogni modulo ha una funzione diversa**
- **Funzioni più importanti:**
  - gestione processore e dei processi
  - gestione memoria (principale e virtuale)

## Gestore dei processi

## Componenti



## **Gestore dei processi**

- **Un sistema operativo con supporto per il multitasking (multiprocessualità) permette di eseguire più programmi contemporaneamente**
- **Sistema Operativo time sharing**
  - Permette la condivisione della CPU tra più processi concorrenti
- **Il tempo di esecuzione del processore è condiviso tra più processi / utenti**
- **Ogni processo in esecuzione ha a disposizione dei quanti di tempo della CPU**

Informatica - A.A. 2011/2012 - Approfondimenti Sistemi operativi

## **Time sharing**

- **Se viene chiesto al sistema di eseguire contemporaneamente due processi A e B, la CPU attuerà un meccanismo di time-sharing → eseguirà per qualche istante (quanto di tempo) il processo A, poi per qualche istante il processo B, poi tornerà ad eseguire il processo A e così via...**
- **Il passaggio dal processo A al processo B e viceversa viene definito cambio di contesto (context switch)**

Informatica - A.A. 2011/2012 - Approfondimenti Sistemi operativi

## Scheduler

- **Le decisioni riguardanti l'esecuzione di un cambio di contesto tra due programmi vengono intraprese da un componente del sistema operativo (software) detto scheduler**
- **Scheduler (da *schedule = mettere in lista*)**
  - Programma che, dato un insieme di richieste di accesso ad una risorsa (processi che contemporaneamente richiedono CPU per essere eseguiti), stabilisce un ordinamento temporale per l'esecuzione di tali richieste

Informatica - A.A. 2011/2012 - Approfondimenti Sistemi operativi

## Scheduling

- **Lo scheduler attua lo scheduling**
  - Ripartizione del tempo di CPU tra tutti i processi concorrenti (i processi attivi nello stesso momento sul calcolatore che si contendono l'uso della CPU)
- **Lo scheduling è un'operazione molto importante per il corretto ed efficiente funzionamento del calcolatore**
- **Consente di migliorare l'utilizzo del processore**

Informatica - A.A. 2011/2012 - Approfondimenti Sistemi operativi

## Algoritmi di scheduling

- **Esistono vari algoritmi di scheduling che tengono conto di varie esigenze e che possono essere più indicati in alcuni contesti piuttosto che in altri**
- **CRITERI da ottimizzare:**
  - *Massimizzare l'utilizzo del processore*
  - *Massimizzare la produttività*
  - *Minimizzare il tempo di completamento*
  - *Minimizzare il tempo di attesa*
  - *Minimizzare il tempo di risposta*
- **NOTA: generalmente si tende ad ottimizzare i valori medi**

Informatica - A.A. 2011/2012 - Approfondimenti Sistemi operativi

## Criteria

1. **Utilizzo del processore:** la CPU deve essere attiva il più possibile (ovvero devono essere ridotti al minimo i possibili tempi morti)
2. **Produttività (throughput):** il numero di processi completati in una determinata quantità di tempo
3. **Tempo di completamento:** il tempo che intercorre tra l'inizio di un processo ed il completamento della sua esecuzione
4. **Tempo d'attesa:** il tempo in cui un processo pronto per l'esecuzione rimane in attesa della CPU (waiting time)
5. **Tempo di risposta:** il tempo che trascorre tra l'inizio del processo e l'ottenimento della prima risposta (dato)

Informatica - A.A. 2011/2012 - Approfondimenti Sistemi operativi

## Round Robin

- Algoritmo di scheduling Round Robin (RR)
- Il più diffuso
- *Esegue i processi nell'ordine d'arrivo assegnando 'a turno' un quanto di tempo di CPU ai processi in attesa*
- Qualora il processo in esecuzione duri più del quanto di tempo assegnato, alla fine del quanto lo si pone alla fine della coda dei processi in attesa e si fa proseguire l'esecuzione al successivo processo in attesa

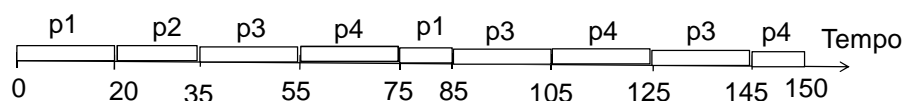
Informatica - A.A. 2011/2012 - Approfondimenti Sistemi operativi

## Esempio

**Ipotesi: i seguenti processi siano in coda con relativa durata in millisecondi, e quanto di tempo stabilito di 20 ms (50 quanti in un secondo):**

***p1 (30) → p2 (15) → p3 (60) → p4 (45)***

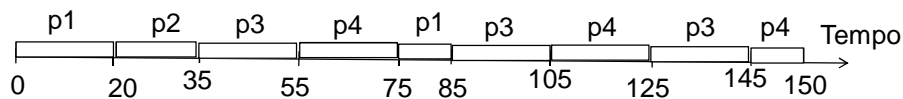
- p1 (20 ms, ne rimangono 10) → p2 (termina la propria esecuzione perché dura meno di 20 ms) → p3 (20 ms, ne rimangono 40) → p4 (20 ms, ne rimangono 25) → p1 (termina perché erano rimasti meno di 20 ms) → p3 (20 ms, ne rimangono 20) → p4 (20 ms, ne rimangono 5) → p3 (termina perché rimanevano 20 ms) → p4 (termina)



Informatica - A.A. 2011/2012 - Approfondimenti Sistemi operativi

## Criterio di ottimizzazione

- Per valutare l'algoritmo di scheduling facciamo riferimento al tempo medio di attesa
- **Tempo di attesa = istante finale di esecuzione – istante di arrivo - durata del processo**



- p1 termina all'istante 85, p2 a 35, p3 a 145, p4 a 150 (tutti arrivati all'istante 0)
- Tempo medio di attesa:  
$$[(85-30)+(35-15)+(145-60)+(150-45)]/4 = 66,25 \text{ ms}$$

Informatica - A.A. 2011/2012 - Approfondimenti Sistemi operativi

## Shortest Next Process First (SNPF)

- SNPF prevede che venga eseguito sempre il processo con il tempo di esecuzione più breve tra quelli in attesa
- **Esempio:** siano sottomessi contemporaneamente i processi dell'esempio precedente, riportati sotto con la rispettiva durata di esecuzione in ms:  
p1 (30) → p2 (15) → p3 (60) → p4 (45)
- I processi vengono eseguiti nel seguente ordine:  
p2 → p1 → p4 → p3
- p2 termina all'istante 15, p1 a 45, p4 a 90, p3 a 150

$$[(15-15)+(45-30)+(90-45)+(150-60)]/4 = 37,5 \text{ ms}$$

Informatica - A.A. 2011/2012 - Approfondimenti Sistemi operativi

## SNPF

- **Si può dimostrare che questo algoritmo è ottimale, in quanto consente di ottenere sempre il valore più basso di tempo d'attesa medio**
- ***Allora perché non è il più usato?***

**Sfortunatamente non è possibile applicarlo, in quanto non è possibile conoscere anticipatamente quanto durerà l'esecuzione del processo**

**Potrebbe essere possibile predirlo, ma solo in determinati contesti**

Informatica - A.A. 2011/2012 - Approfondimenti Sistemi operativi

## Scheduling con priorità

- **Una priorità (numero intero) è assegnata ad ogni processo**
- **La CPU è assegnata al processo con più alta priorità**
  - es: il più piccolo intero = la più alta priorità
- **Possibile uso di preemption (o pre-rilascio): un processo in esecuzione viene interrotto, con l'intenzione di ripristinarlo in un secondo momento, per dare spazio ad un altro processo a priorità più alta**
  - **Preemptive**
  - **Non preemptive**

Informatica - A.A. 2011/2012 - Approfondimenti Sistemi operativi



## Scheduling con priorità

- **Problema: *Starvation*** – i processi a più bassa priorità potrebbero non essere mai eseguiti (attesa infinita)
- **Soluzione: *Aging*** – al trascorrere del tempo di attesa si incrementa la priorità di un processo che attende

Informatica - A.A. 2011/2012 - Approfondimenti Sistemi operativi

## Scheduling per multiprocessori

- **Lo scheduling nei sistemi multiprocessore è più complesso**
- **Si possono avere processori omogenei (tutti uguali) o disomogenei (processori diversi)**
- **Problema del bilanciamento del carico (*load balancing*)**

Informatica - A.A. 2011/2012 - Approfondimenti Sistemi operativi

## **Gestore della memoria centrale**

### **Gestione della memoria centrale**

- **La gestione concorrente di molti processi comporta la presenza di molti programmi in memoria centrale**
  - ***la memoria centrale diventa una risorsa condivisa da suddividere tra i vari programmi***
    - *Risorsa in generale scarsa: i programmi e i file possono eccedere la memoria centrale disponibile*
- **Il gestore della memoria applica tecniche per gestire il conflitto tra dimensione della memoria centrale disponibile e spazio complessivamente richiesto da programmi e dati che devono essere gestiti in modo concorrente**

## Come gestire la “scarsità” di memoria

- **Strategia seguita:**

- Mantenere in memoria centrale solo una porzione dei programmi in esecuzione e dei relativi dati (la porzione necessaria in quel momento)
- Alternare le porzioni dei programmi mantenute in memoria centrale: i programmi (o le loro porzioni) possono essere ‘caricati’ e ‘scaricati’ dalla memoria centrale durante l’esecuzione a seconda della necessità

- **Si basa sul concetto di località spazio-temporale**

- Non vale la pena di caricare in memoria centrale tutto il codice di un programma perché l’esecuzione in un dato istante si limita ad una porzione del codice stesso, che spesso viene rieseguita
- Principio del 10-90: il 10% del codice spesso copre il 90% dell’esecuzione

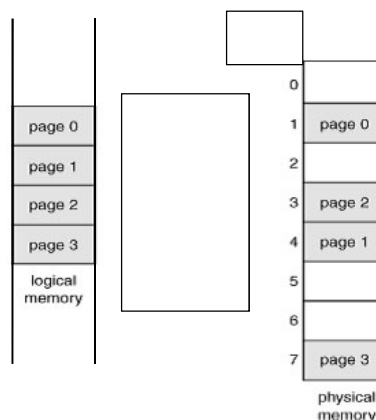
Informatica - A.A. 2011/2012 - Approfondimenti Sistemi operativi

## Spazi di indirizzamento

- **Necessità di considerare due spazi di indirizzamento diverso per ogni programma:**

**Logico:** intervallo di celle contigue che partono dall’indirizzo 0 (iniziale) in cui si immagina siano collocate le istruzioni del programma

**Fisico:** lo spazio di memoria centrale in cui risiede effettivamente il codice di un programma



Informatica - A.A. 2011/2012 - Approfondimenti Sistemi operativi

## Paginazione

- **Frammentazione della memoria in blocchi di dimensione prefissata detti pagine**
- **Lo spazio di indirizzamento logico del programma è suddiviso in sezioni, di dimensioni fisse e uguali tra loro, dette pagine logiche**
- **Lo spazio fisico di indirizzamento presente nel calcolatore è anch'esso diviso in pagine fisiche, della stessa dimensione delle pagine logiche**

Informatica - A.A. 2011/2012 - Approfondimenti Sistemi operativi

## Vantaggi

- **Grazie alla paginazione è possibile:**
  - Tenere in memoria solo la porzione ridotta del programma che è necessaria al momento dell'utilizzo
  - Estendere la dimensione di un processo utilizzando spazi di memoria non necessariamente contigui

Informatica - A.A. 2011/2012 - Approfondimenti Sistemi operativi

## Meccanismo di base

- Solo alcune pagine logiche del programma in esecuzione vengono caricate in (altrettante – stessa dimensione) pagine fisiche della memoria centrale
- Le pagine logiche necessarie vengono caricate di volta in volta in base all'esigenza
- *Quante pagine di un programma caricare e quante/quali pagine sostituire dipende dalla politica adottata*

Informatica - A.A. 2011/2012 - Approfondimenti Sistemi operativi

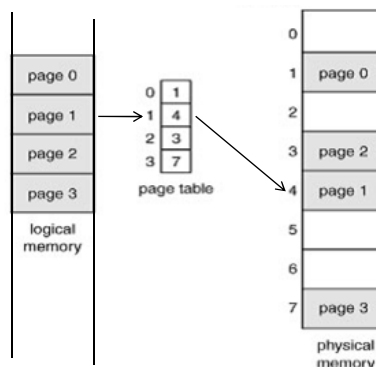
## Memoria virtuale

- Nel caso della paginazione, il gestore della memoria offre ai programmi in esecuzione la visione di una *memoria virtuale*
- La memoria virtuale  $V$  può essere maggiore di quella fisica  $P$ , cioè composta da più pagine di quelle che possono stare contemporaneamente in  $P$  (quelle non caricate in memoria centrale rimangono disponibili in memoria di massa)

Informatica - A.A. 2011/2012 - Approfondimenti Sistemi operativi

## Conseguenza

- I programmi con dimensione  $S > P$  ( $P$  = spazio fisico di memoria centrale) possono essere eseguiti se  $S < V$  ( $V$  = spazio di memoria virtuale)
- E' il gestore della memoria che si occupa di mantenere la conversione degli indirizzi da logici a fisici
  - Si usa una tabella di conversione



Informatica - A.A. 2011/2012 - Approfondimenti Sistemi operativi

## Swapping

- Un modo tipico di gestire la memoria virtuale consiste nell'abilitare il sistema operativo a trasferire il contenuto dalla memoria centrale ad un'area della memoria di massa, detta "area di swap"
- Lo swapping viene applicato trasferendo su disco le informazioni relative ai processi in stato di "attesa"
  - Attesa di una periferica di Input/Output
  - Attesa della CPU

Informatica - A.A. 2011/2012 - Approfondimenti Sistemi operativi

## Processi e memoria

- **La gestione della memoria deve essere coordinata con la gestione dei processi:**
  - le pagine di programma che sono attualmente in esecuzione o che contengono dati attualmente utilizzati devono risiedere in memoria centrale
  - se una pagina necessaria ad un processo in esecuzione non si trova in memoria centrale (page fault), il processo viene sospeso e passa in stato di attesa
  - dopo l'operazione di input da memoria di massa della pagina necessaria il processo torna ad essere pronto per l'esecuzione