

SCHEDULING CPU > S.1

Si intende il ^{Policy?} meccanismo di esecuzione dei programmi nei sistemi con basso carico di risorse.

CPU/I/O BURST CYCLE S.1.1

È un modo di descrivere il flusso di esecuzione del processo come sequenze di esecuzione CPU (CPU burst) e seq. di esecuzione I/O (I/O burst).

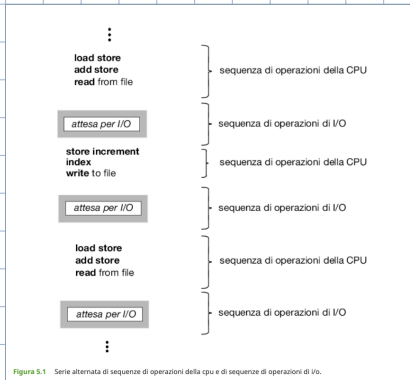


Figura 5.1 Serie alternata di sequenze di operazioni della CPU e di sequenze di operazioni di I/O.

Al termine delle quali ci sarà la terminazione del processo.

Durante la I/O burst il processo si mette in attesa.

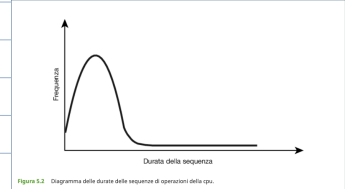


Figura 5.2 Diagramma della durata delle sequenze di operazioni della CPU.

Il short-term scheduler ha il compito di designare il prossimo processo della ready queue. La queue a sua volta può essere implementata in diversi modi ed è formata dai PCB dei processi.

DECISIONI DELLO SCHEDULER

La policy dello scheduler può entrare in atto quando:

1. Proc. passa running \Rightarrow waiting (es: per I/O / wait();)
2. // // running \Rightarrow ready (es: interrupt)
3. // // waiting \Rightarrow ready (es: I/O ha finito)
4. Processo termina

Da cui si possono dividere in:

- Gruppo 1,4: Non preemptive / senza prelazione
Richiedono la scelta di un nuovo processo.
- Gruppo 2,3: Preemptive / con prelazione

La scelta del processo può portare a race condition.

Anche il kernel può essere progettato con o senza prelazione. (senza = aspetta che il processo vada in archit.)

Se è con prelazione deve implementare dei lock sui dati per evitare race condition.

DISPATCHER > S.1.4

È un modulo che implementa la policy di scheduling ovvero: context switch, passaggio mod. user, resume del programma.

Il tempo impiegato per le 3 attività è la latenza di dispatch.



Figure 5.11 Switch del Dispatch

CRITERI ALGO. SCHEDULING > S.2

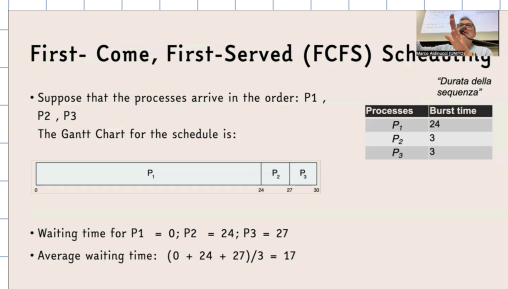
Sono:

- Uso CPU: Il più possibile.
- Throughput: Il più possibile processi per unità di tempo.
- Turnaround Time: Il tempo per eseguire il processo.
- Waiting Time: Il tempo di attesa nella ready queue.
- Response Time: Tempo tra la richiesta e la prima risposta.

FCFS SCHEDULING > S.3.1

L'algoritmo First Come First Serve consente lo scheduling dei processi.

È una coda FIFO in cui il primo che arriva viene servito nella CPU.



• Vantaggi:

- facile realizzazione e facile meccanismo.

• Svantaggi:

- Generalmente non è il più veloce perché dipende dall'ordine di arrivo e dalla durata dei processi.

Quindi, ha un average waiting time alto.

L'algoritmo FCFS è senza prelazione