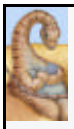
Capitolo 5: Scheduling della CPU



Capitolo 5: Scheduling della CPU

- Concetti fondamentali
- Criteri di scheduling
- Algoritmi di scheduling
- Scheduling per sistemi multiprocessore
- Scheduling real-time
- Scheduling dei thread
- Esempi di sistemi operativi
- Scheduling dei thread in Java
- Valutazione degli algoritmi

Serie alternata di sequenze di operazioni della CPU e di sequenze di operazioni di I/O

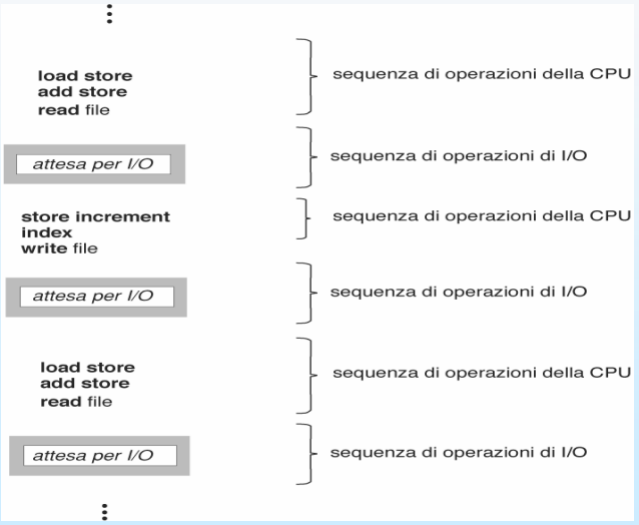
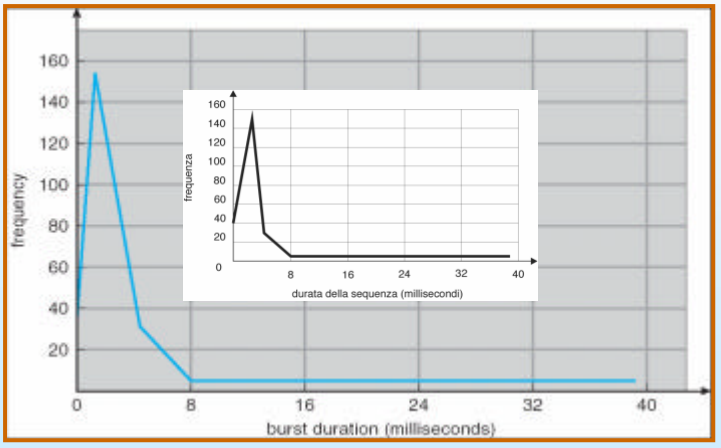
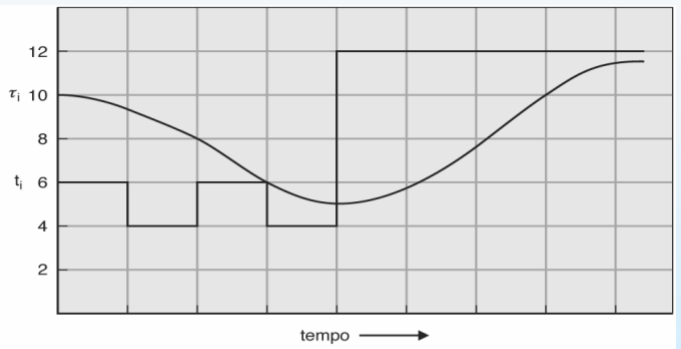


Diagramma delle durate delle sequenze di operazioni della CPU

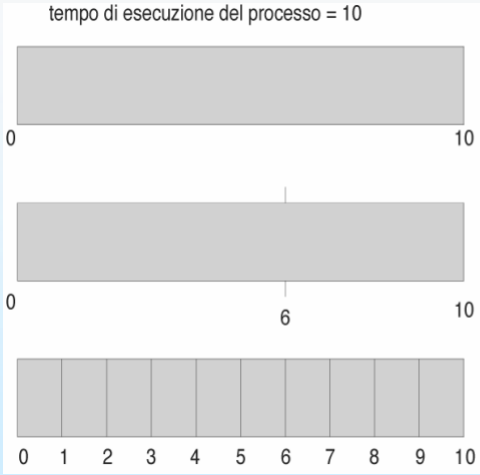


Predizione della lunghezza della successiva sequenza di operazioni della CPU (*CPU burst*)



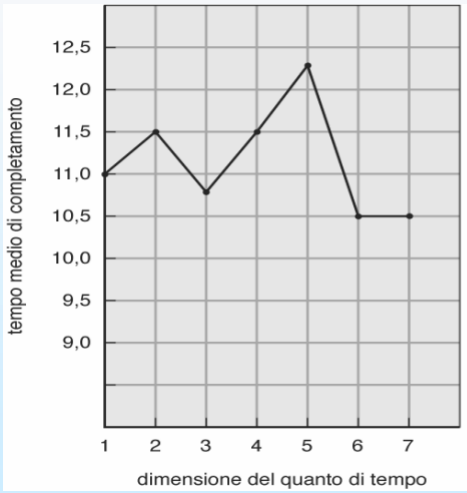
CPU burst (t_i)	6	4	6	4	13	13	13	...
predizione (τ_i)	10	8	6	6	5	9	11	12	...

Quanto di tempo e cambio di contesto



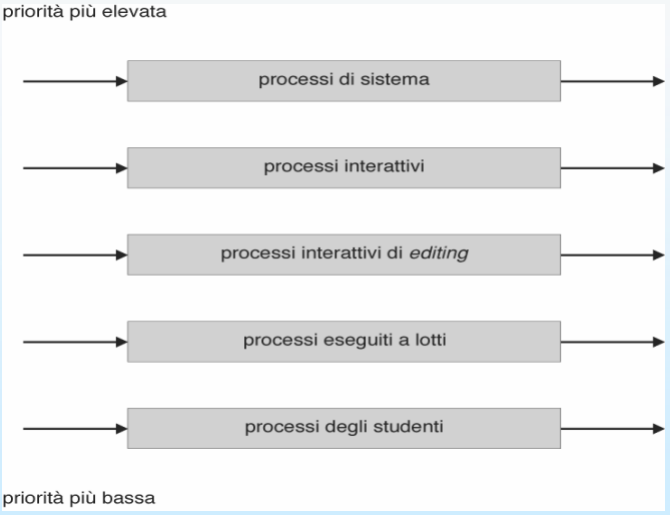
lunghezza del quanto di tempo	cambio di contesto
12	0
6	1
1	9

Variazione del tempo di completamento in funzione del quanto di tempo

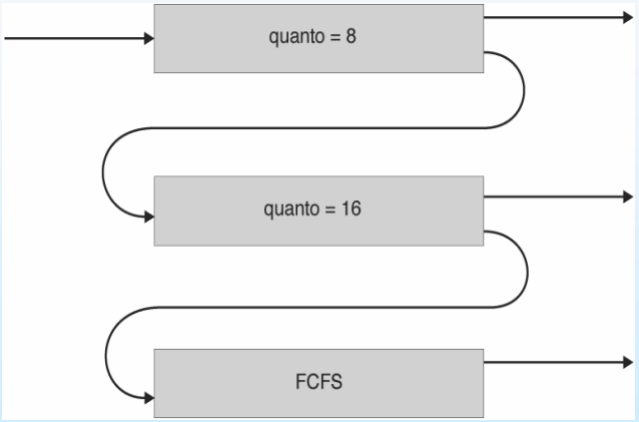


processo	tempo
P_1	6
P_2	3
P_3	1
P_4	7

Scheduling a code multiple



Code multiple con retroazione



API Pthread per lo scheduling

```
#include <pthread.h>
#include <stdio.h>
#define NUM_THREADS 5
int main(int argc, char *argv[ ])
{
    int i;
   pthread_t tid[NUM_THREADS];
   pthread_attr t_attr;
   /* ottiene gli attributi di default */
   pthread_attr_init(&attr);
   /* imposta l'algoritmo di scheduling a PCS o SCS */
   pthread_attr_setscope(&attr, PTHREAD_SCOPE_SYSTEM);
   /* imposta I criteri di scheduling - FIFO, RT, oppure ALTRO
   */
   pthread_attr_setschedpolicy(&attr, SCHED OTHER);
   /* genera i thread */
   for (i = 0; i < NUM_THREADS; i++)
        pthread_create(&tid[i],&attr,runner,NULL);
```

API Pthread per lo scheduling

```
   /* adesso aspetta la terminazione di
      tutti i thread */
   for (i = 0; i < NUM_THREADS; i++)
        pthread_join(tid[i], NULL);
}
 /* Ogni thread inizia l'esecuzione da
    questa funzione */
void *runner(void *param)
{
   printf("I am a thread\n");
   pthread_exit(0);
}
```


Scheduling di Solaris

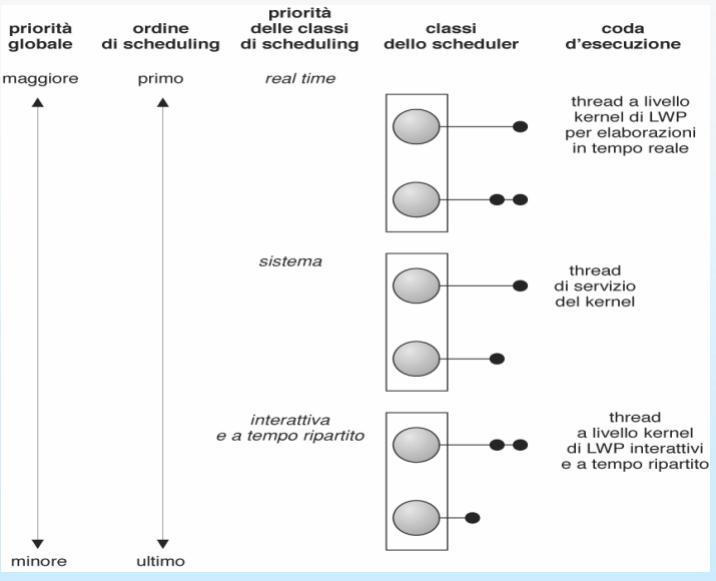


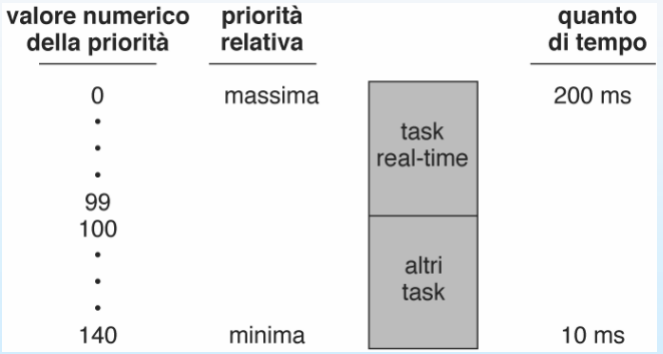
Tabella di dispatch di Solaris

priorità	quanto di tempo	quanto di tempo esaurito	ripresa dell'attività
0	200	0	50
5	200	0	50
10	160	0	51
15	160	5	51
20	120	10	52
25	120	15	52
30	80	20	53
35	80	25	54
40	40	30	55
45	40	35	56
50	40	40	58
55	40	45	58
59	20	49	59

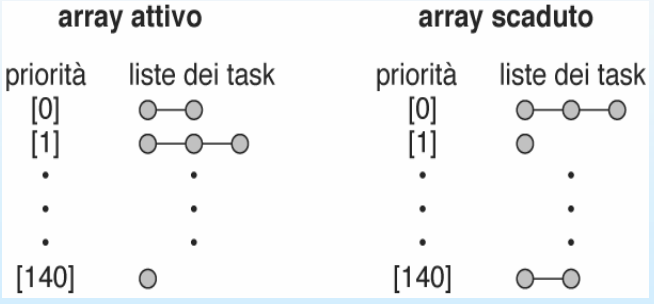
Priorità di Windows XP

	real-time	high	above normal	normal	below normal	idle priority
time-critical	31	15	15	15	15	15
highest	26	15	12	10	8	6
above normal	25	14	11	9	7	5
normal	24	13	10	8	6	4
below normal	23	12	9	7	5	3
lowest	22	11	8	6	4	2
idle	16	1	1	1	1	1

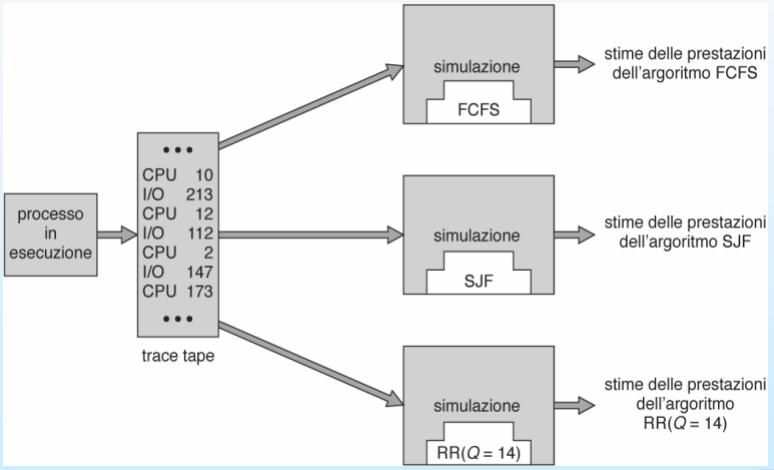
Relazione fra le priorità e la lunghezza del quanto di tempo



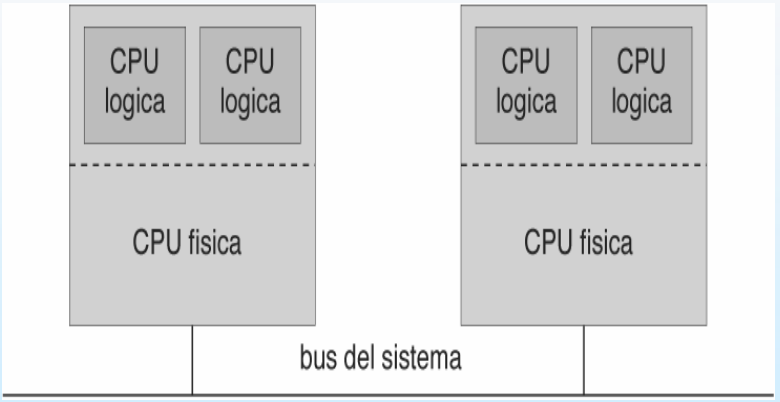
Liste dei task indicizzate sulla base delle priorità



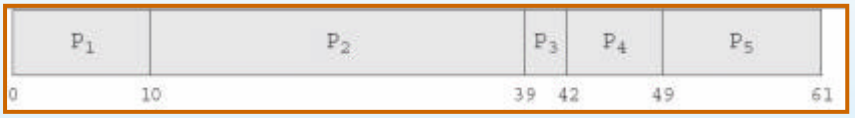
Valutazione di algoritmi di scheduling della CPU tramite una simulazione



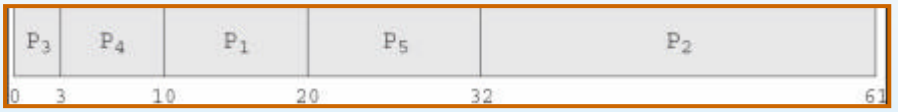
Tipica architettura SMT



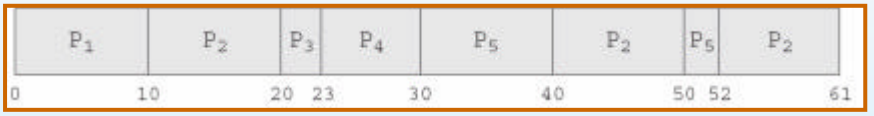
Con l'algoritmo FCFS i processi si eseguono secondo lo schema seguente



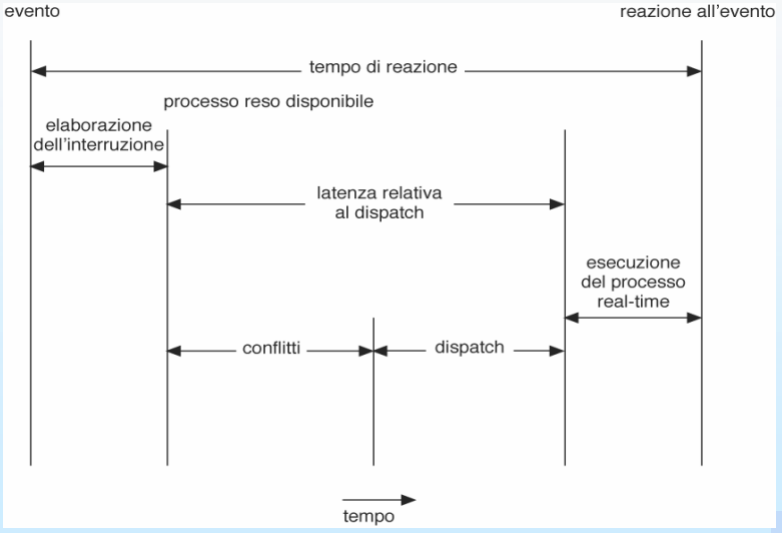
Con l'algoritmo SJF senza prelazione i processi si eseguono come segue



Con l'algoritmo RR i processi si eseguono come segue



Latenza relativa al dispatch



Fine del Capitolo 5