

Sistemi Operativi

Laurea in Ingegneria Informatica

Università Roma Tre

Docente: Romolo Marotta

CPU scheduling

Tipi di scheduling

Tramite l'analisi degli stati di un processo, si sono individuati tre classi di CPU scheduling:

- Long-term scheduling
 - Ammette nuovi processi al sistema
 - Controlla il livello di multiprogrammazione
 - Dipende dal carico del sistema
- Mid-term scheduling
 - Swap in/swap out di processi
 - Dipende dal livello di multiprogrammazione e memoria disponibile
- Short-term scheduling
 - Ammette processi ad andare in esecuzione sul processore
 - Invocato di frequente

Metriche per decisioni di scheduling

Lo scheduler può adottare diverse politiche di scheduling

Come comparare gli effetti delle politiche di scheduling?

- Uso di criteri per catturare specifici aspetti di interesse
 - Prestazionali
 - Non prestazionali
- Soggetti interessati
 - Sistema
 - Utente

Metriche per decisioni di scheduling

- Criteri **prestazionali** orientati **all'utente**
 - Tempo di risposta: tempo necessario affinché un processo inizi a produrre l'output
 - Tempo di turnaround: tempo totale tra la creazione di un processo ed il suo completamento
 - Deadline: percentuale di specifiche scadenze temporali rispettate
- Criteri **non prestazionali** orientati **all'utente**
 - Prevedibilità: variazione ridotta del tempo di risposta o di turnaround

Metriche per decisioni di scheduling

- Criteri **prestazionali** orientati al sistema
 - Throughput: numero di processi completati per unità di tempo
 - Utilizzazione: percentuale di tempo in cui una determinata risorsa risulta impegnata
- Criteri **non prestazionali** orientati al sistema
 - Fairness: evitare starvation di processi
 - Priorità: favorire processi con maggior priorità
 - Bilanciamento delle risorse: evitare il sottoutilizzo di risorse e favorire processi che non richiedono risorse in sovraccarico

Preemptive e non-preemptive scheduling

Non-preemptive scheduling:

- Una azione di scheduling **attende** che il processo abbandoni la cpu
 - Processo terminato
 - Bloccato per I/O

Preemptive scheduling:

- Una azione di scheduling può attendere che il processo abbandoni la CPU o **interrompere** il processo in esecuzione indipendentemente dalle attività che esso sta svolgendo
 - All'occorrenza di un interrupt
 - All'occorrenza di un evento (e.g., terminazione di I/O)
 - Invocazione di syscall

First Come First Serve (FCFS)

- Non-preemptive
- I processi in stato ready vengono eseguiti nell'ordini in cui sono stati inseriti nella ready-to-run queue

Pros:

- Semplice
- No starvation

Cons:

- Non garantisce minimo turnaround time medio o waiting time medio
- Soggetto ad una forte variabilità
- Non massimizza l'utilizzo delle risorse

Processo	Tempo di CPU
P1	2
P2	6
P3	1



$$\text{AVG turnaround} = (1 + 3 + 9)/3 = 4.33$$

$$\text{AVG waiting time} = (0 + 1 + 3)/3 = 1.33$$



$$\text{AVG turnaround} = (6 + 8 + 9)/3 = 7.66$$

$$\text{AVG waiting time} = (0 + 6 + 8)/3 = 4.66$$

First Come First Serve (FCFS)

CPU

I/O

CPU-bound:

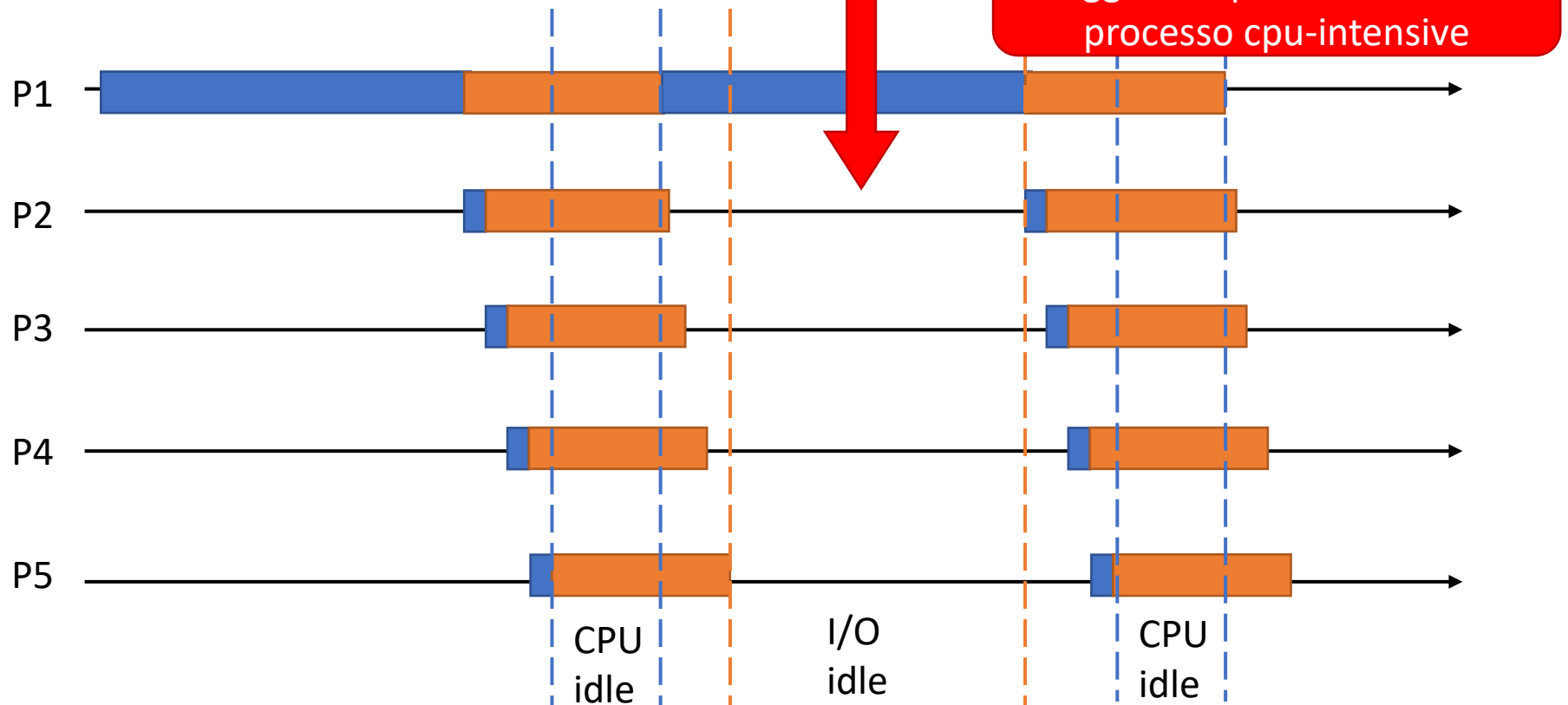
- P1

I/O bound:

- P2,P3,P4,P5

Arrival:

P1,P2,P3,P4,P5



Shortest Job First (SJF)

- Non-preemptive
- I processi vengono schedulati in accordo a quanto tempo occuperanno la CPU
- Anche chiamato Shortest Process First (SPF) o Shortest Next CPU-burst First

Pros:

- Minimizza turnaround/waiting time
- Favorisce il throughput

Cons:

- Possibilità di starvation
- Necessità di predire per quanto tempo un processo riesce in CPU

Può supportare prelazione (Shortest Remaining Time Next - SRTN):

- L'arrivo di un processo ready con minor lunghezza stimata di CPU-burst

Processo	Tempo di CPU
P1	2
P2	6
P3	1



$$\text{AVG turnaround} = (1 + 3 + 9)/3 = 4.33$$

$$\text{AVG waiting time} = (0 + 1 + 3)/3 = 1.33$$

$$\text{Avg: } S_{n+1} = \frac{1}{n} \sum_{i=0}^n T_i$$

Exp Avg: $S_{n+1} = \alpha T_n + (1 - \alpha) S_n$
 α vicino ad 1 => maggior peso ad osservazioni recenti => maggior instabilità

Round Robin (RR)

- Preemptive
- Ai viene assegnato un quanto di tempo (time quantum o time slice)
- Anche chiamato Shortest Process First (SPF) o Shortest Next CPU-burst First

Pros:

- No starvation
- Attesa massima limitata: $(n - 1)q$

Cons:

- La taglia della time slice è critica per le performance
- Unfair: sfavorisce processi I/O bound

Processo	Tempo di CPU
P1	2
P2	6
P3	1



$$\text{AVG turnaround} = (6 + 8 + 9)/3 = 7.66$$

$$\text{AVG waiting time} = (0 + 6 + 8)/3 = 4.66$$



$$Q = 2$$

$$\text{AVG turnaround} = (9 + 4 + 8)/3 = 7$$

$$\text{AVG waiting time} = (3 + 2 + 2)/3 = 2.33$$

Round Robin (RR)

La taglia della time slice è critica per le performance

- Q maggiore del massimo CPU burst \Rightarrow RR collassa su FIFO
- Q minore o uguale al tempo di context-switch
 - \Rightarrow Il processore spende più del 50% del suo tempo ad eseguire context-switch
- Q minore del tempo necessario a compiere un'unità di lavoro (e.g. attivazione di I/O)
 - \Rightarrow Maggiori tempi di attesa e sottoutilizzo dispositivi di I/O

Processo	Tempo di CPU
P1	2
P2	6
P3	1



Q = 6

AVG turnaround = $(6 + 8 + 9)/3 = 7.66$

AVG waiting time = $(0 + 6 + 8)/3 = 4.66$



Q = 2

AVG turnaround = $(9 + 4 + 8)/3 = 7$

AVG waiting time = $(3 + 2 + 2)/3 = 2.33$

Round Robin (RR)

Unfair: sfavorisce processi I/O bound

- Processi I/O bound non utilizzano tutto il loro quanto
- Processi CPU bound tendono a rientrare immediatamente nella ready queue

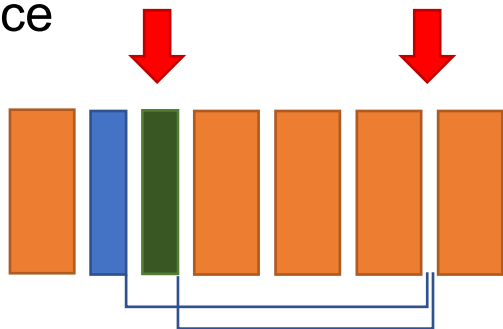
Soluzione: Virtual Round Robin (VRR)

- I processi hanno un credito per non aver speso tutto il quanto
 - I processi ready con credito hanno una coda ausiliaria dedicata
 - Lo scheduler sceglie il processo prima dalla coda ausiliaria e poi dalla ready-to-run queue
 - Un processo proveniente dalla coda ausiliaria esegue al più per un tempo pari al suo credito, ossia il tempo non utilizzato della precedente time slice
- $Q = 1$
 - P2 CPU-bound, P1 e P3 I/O bound

Processo	Tempo di CPU
P1	0.5
P2	6
P3	0.5



RR



VRR



Priority Scheduling

- Una priorità viene assegnata ai processi
- Il dominio delle priorità è un insieme su cui esiste un ordinamento totale
- I processi vengono schedulati secondo priorità decrescenti (prima i processi a priorità più alta)
- Non è necessario che priorità più alte abbiano valori maggiori rispetto a priorità più basse
 - Esempio
 - Dominio: numeri interi positivi incluso lo 0
 - Priorità 0 = priorità massima
- Tipicamente processi a medesima priorità vengono serviti con politica FCFS o RR

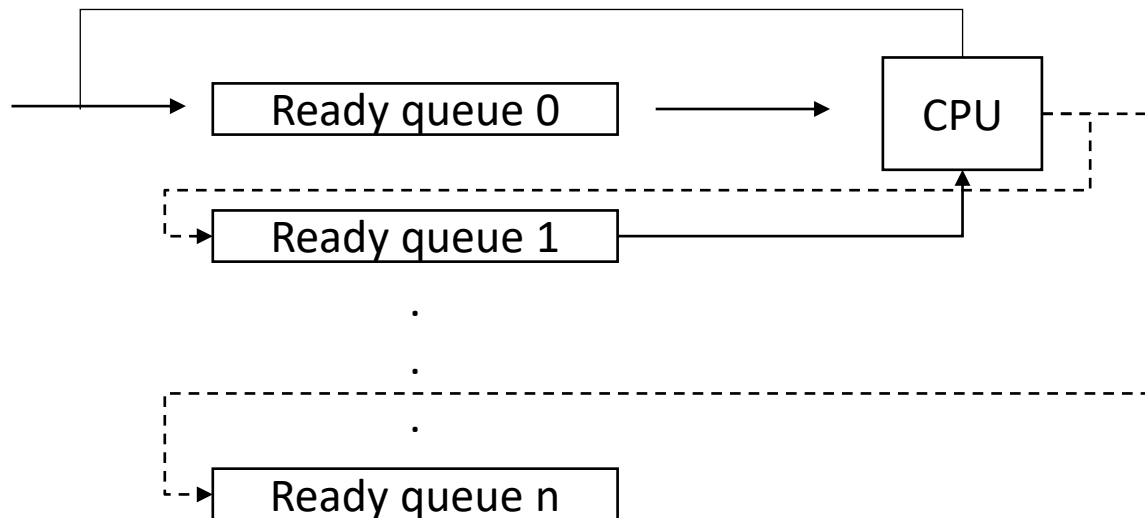
Priority Scheduling

FCFS, SJF, SRTN sono specifici schemi a priorità:

- FCFS
 - Dominio: tempo
 - Assegnazione: istante di tempo di inserimento in coda
 - Selezione: processo con valore di priorità minore
- SJF
 - Dominio: tempo
 - Assegnazione: durata di CPU-burst
 - Selezione: processo con valore di priorità minore
- SRTN
 - Dominio: tempo
 - Assegnazione: durata di CPU-burst residua
 - Selezione: processo con valore di priorità minore

Multilevel Feedback Queue

- Più code FCFS o RR
- Ad ogni coda è associata una priorità
- Si schedulano processi in una coda a priorità più bassa quando tutte le code a priorità più alta sono vuote
- Se un processo utilizza tutto il suo quanto verrà spostato in una coda con minor priorità
- Processi a priorità bassa possono soffrire di starvation
 - Problema parzialmente alleviato se la time slice è crescente per priorità decrescenti



Priority Scheduling

Processi a priorità bassa possono soffrire di starvation

- Dipende da come sono definite ed assegnate le priorità
 - Vedi SJF, SRTN, MFQ
- Meccanismi di **aging**, ossia di invecchiamento, per far salire le priorità di processi a bassa priorità (CPU-bound)
- Highest Response Ratio Next (HRRN)
 - Dominio: Reali positivi
 - Assegnazione: $p = \frac{w+s}{s}$ dove w è il tempo speso in coda e s è la previsione di durata del prossimo CPU-burst
 - Selezione: $\max(p)$