

Programma di Scheduling: Manuale d'uso

Francesco Alban

1 Introduzione

Questo programma, realizzato in Java, è in grado di effettuare lo scheduling dei processi, ed è in grado di schedulare secondo questi algoritmi:

- **FCFS**
- **SJF-non preemptive**
- **SJF-preemptive**
- **HRRN non preemptive**
- **Priorità non preemptive**
- **Priorità preemptive**
- **Round Robin attesa**
- **Round Robin risposta**

È composto da 4 finestre, una per la scelta del numero dei processi (figura 1), una per l'inserimento dati e la scelta dell'algoritmo (figura 2), una per la visualizzazione del grafico di esecuzione dei processi (figura 3) e l'ultima per la visualizzazione delle statistiche dei processi (figura 4).

2 Specifiche tecniche

Questo programma può schedulare fino a un massimo di 10 processi. Gli algoritmi *SJF-preemptive*, *Priorità preemptive*, *Round Robin attesa*, *Round Robin risposta* possono schedulare processi con valori di tempo di arrivo e tempo di computazione (burst) non inferiori al decimo mentre gli altri algoritmi supportano valori al centesimo.

2.1 Algoritmi a Priorità

La priorità negli algoritmi a priorità preemptive e non preemptive è maggiore tanto più il valore è basso. Sono supportate anche priorità con valori negativi.

Ad esempio il processo con priorità 0 ha una priorità maggiore di un processo con priorità 1.

2.2 Algoritmi Round Robin

Prima sono stati elencati due tipi di algoritmi di Round Robin: *risposta* e *attesa*. Questi due algoritmi differenziano nel modo in cui i processi nuovi vengono inseriti in coda, in modo da minimizzare, rispettivamente il tempo di risposta o il tempo di attesa dei processi. Nel Round Robin risposta i processi nuovi vengono inseriti in testa alla coda mentre in quello a modalità attesa i processi nuovi vengono messi in ultima posizione in coda. Ad esempio se ci sono due processi con questi valori:

<i>Processo</i>	<i>Tempo di arrivo</i>	<i>Tempo di computazione</i>
1	0	1
2	1	1

Al tempo 0 ovviamente andrà in esecuzione il processo 1. Al tempo 1 nel Round Robin risposta andrà in esecuzione il processo 2 mentre nel Round Robin attesa andrà in esecuzione il processo 1.

3 Contenuto del file zip

Il file zip contiene i seguenti file e cartelle:

- la cartella *src* contenente i file sorgenti *.java*;
- la cartella *classFile* contenente i file *class*;
- il file *Manuale.pdf*.

4 Lancio del programma

Per lanciare il programma, bisogna essere muniti di *java* e si deve dare il seguente comando:

```
java classFile.TestGui
```

e verrà visualizzata la finestra in figura 1 (lanciare il comando all'esterno della cartella *classFile*).

La risoluzione minima consigliata per la corretta visualizzazione delle finestre del programma è di 1024x768 poichè la dimensione massima delle finestre è quest'ultima.

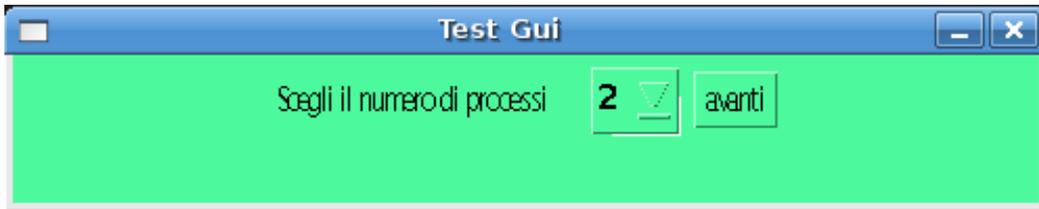


Figura 1: Finestra per la scelta dei numero dei processi



Figura 2: Finestra per l'inserimento dei dati e la scelta dell'algoritmo



Figura 3: Finestra per la visualizzazione del grafico di esecuzione dei processi

Statistiche Processi			
	Visualizza grafico	Cambia algoritmo	Cambia numero processi
Processo numero	Tempi di attesa	Tempi di risposta	Tempi di turnaround
1	0.0	0.0	1.0
2	3.0	0.0	8.0
3	0.0	0.0	1.0
4	15.0	15.0	22.0
5	0.0	0.0	2.0
6	19.0	19.0	27.0
7	4.0	4.0	6.0
8	0.0	0.0	1.0
9	0.0	0.0	1.0
10	2.0	2.0	7.0
	Tempo di attesa medio	Tempo di risposta medio	Tempo di turnaround medio
	4.3	4.0	7.6

Figura 4: Finestra per la visualizzazione delle statistiche dei processi