

Risolutore di puzzle – Parte 2

Programmazione concorrente e distribuita
Progetto A.A. 2014/2015

1 Descrizione del progetto

La parte 2 del progetto consiste nel modificare il programma realizzato per la parte 1 in modo tale che la logica dell'algoritmo di ricostruzione del puzzle sia concorrente.

In particolare, l'input e l'output del nuovo programma rimangono identici: l'input consiste di un file contenente l'elenco completo dei pezzi mischiati, e l'output di un file contenente il testo del puzzle, il puzzle ricostruito in forma tabellare e le sue dimensioni.

I dettagli su come implementare la logica concorrente di risoluzione del puzzle sono lasciati allo studente, che dovrà darne spiegazione nella relazione allegata. È richiesto unicamente che ci siano più attività concorrenti che ricostruiscono il puzzle, che i vari thread coinvolti abbiano un carico di lavoro uniforme e che non tengano occupata la CPU inutilmente. Vanno inoltre evitati problemi di interferenza e di deadlock.

2 Requisiti obbligatori

La parte 2 del progetto deve soddisfare i seguenti requisiti obbligatori:

- L'input e l'output del programma avranno le stesse specifiche definite nella parte 1 del progetto.
- Il programma non avrà un'interfaccia grafica.
- Allegato al programma ci sarà una relazione.
- Allegato al programma ci sarà un **makefile**
- Allegato al programma ci sarà uno **script Bash** che si chiamerà **puzzlesolver.sh** nella root del progetto. Lo script accetterà due parametri: `nome_file_input` `nome_file_output`, rispettivamente il nome del file di input e quello di output. Lo script eseguirà il comando di avvio del programma Java passandogli i due parametri.

3 Valutazione del progetto - Relazione

Al fine di permettere la valutazione del progetto è richiesto di consegnare, oltre al programma, una **breve relazione** che illustri in modo **preciso e sintetico**:

- una breve spiegazione della logica dell'algoritmo *parallelo* di ricostruzione del puzzle;
- quanti/quali thread vengono avviati dal programma, e quanti/quali di questi thread possono essere attivi concorrentemente;

- quali costrutti di concorrenza di Java sono stati utilizzati (**synchronized**, **wait()** e **notify()**, gestione di **InterruptedException**, **interrupt()**, **interrupted()**, **join()**...) e il loro utilizzo nel contesto del programma;
- cosa assicura che in ogni esecuzione del programma non ci sono interferenze, né deadlock, né thread che fanno inutilmente attesa attiva;
- i cambiamenti effettuati rispetto la parte 1: cosa è cambiato nel codice e nell'organizzazione delle classi e perché?

NOTA: La valutazione del progetto dipenderà in **ugual misura dalla qualità della relazione e dalla correttezza e qualità del codice**. Si osserva inoltre che la valutazione del progetto terrà conto della capacità di risolvere il problema assegnato in modo semplice ma esauriente. Sono quindi preferibili i progetti che implementano in modo semplice e chiaro una soluzione corretta, mentre viene **SCORAGGIATA l'aggiunta di funzionalità e aspetti grafici non richiesti dalla presente specifica**. Anche in questa fase la valutazione del codice terrà conto dell'aderenza ai principi basilari della programmazione ad oggetti in Java.

4 Regole per la consegna del progetto

Il progetto dovrà essere realizzato da ogni singolo studente in modo **INDIPENDENTE**.

4.1 Come verrà effettuato il test operativo del progetto

Per quanto riguarda la valutazione del progetto, questo verrà eseguito localmente su un sistema operativo Linux, dove vi è una installazione di Java 7. I computer dei laboratori attualmente hanno per default la versione 6 di Java, ma è possibile usare la versione 7 personalizzando l'ambiente di esecuzione come indicato alla pagina

<http://www.studenti.math.unipd.it/index.php?id=corsi#c620>.

Al fine di standardizzare la procedura di valutazione il programma dovrà aderire a quanto specificato nella sezione "Requisiti obbligatori".

ATTENZIONE: se un progetto non compila o non soddisfa i requisiti obbligatori sarà considerato insufficiente.

4.2 Cosa consegnare

La cartella principale del pacchetto da consegnare, dovrà essere chiamata "parte2". Nel caso di consegna contemporanea di più parti, il pacchetto avrà più cartelle, una per ciascuna parte.

4.3 Come e quando consegnare

Il progetto va consegnato dalle macchine del laboratorio invocando il comando

```
consegna programmazione3-14-15
```

dalla directory contenente i file da consegnare. **Non** saranno accettate altre modalità di consegna (ad es. via email). È possibile consegnare remotamente il progetto usando il server `ssh.studenti.math.unipd.it`.

Le date di consegna del progetto saranno indicate sul sito del corso.