

Programmazione I

Prova di Programmazione – 18 gennaio 2011 – 2 ore 30 min

Partendo dal frammento di codice fornito, scrivere un programma di gestione dell'accesso ad un dispositivo a disco, in cui il disco è modellato come una sequenza di *blocchi* numerati a partire da 1. Il programma gestisce l'inserimento ed il servizio di *richieste*, dove una richiesta è costituita da una sequenza di blocchi del disco da leggere. La sequenza è in generale non contigua, ed è ordinata per indice di blocco crescente. Il numero massimo di elementi di una sequenza è definito a tempo di scrittura del programma. All'inizio non vi è alcuna richiesta in attesa di servizio.

1. **inserisci_richiesta** Inserisce una richiesta da servire. Se il numero di richieste pendenti, cioè ancora da servire, è già uguale ad un valore massimo definito a tempo di scrittura del programma, l'operazione fallisce. Altrimenti si chiede all'utente di inserire la sequenza (da *stdin*), cioè l'indice dei blocchi a cui accedere. L'utente non è obbligato ad inserire gli indici in ordine crescente, è invece il programma che garantisce automaticamente l'ordinamento.
2. **stampa_richieste** Stampa, per ciascuna delle richieste pendenti, la sequenza di indici di blocchi a cui accedere. Il formato è quello illustrato dal seguente esempio di stampa, a sua volta relativo allo stato in cui ci si troverebbe dopo l'inserimento di due richieste, una da tre blocchi e l'altra da due blocchi, rispettivamente:
21 32 42
10 50
3. **salva_stato** Salva lo stato corrente delle richieste in un file di testo dal nome definito a tempo di scrittura del programma.
4. **carica_stato** Ricarica lo stato dal file. L'eventuale stato precedente è perso.
5. **servi_richiesta** Serve la richiesta il cui primo blocco da leggere abbia indice più vicino all'indice dell'ultimo blocco letto. Se non è ancora stata servita alcuna richiesta, serve quella il cui primo blocco ha indice minimo. La richiesta servita viene eliminata dall'elenco delle richieste pendenti. Per esempio, a partire dallo stato illustrato nell'esempio al punto 2 e supponendo che non sia stata ancora servita alcuna richiesta, verrebbe servita la seconda richiesta, e l'indice dell'ultimo blocco acceduto diverrebbe 50. A questo punto, se si inserisse un'ulteriore richiesta i cui blocchi avessero indici 60 100 200, si avrebbero le richieste
21 32 42
60 100 200
e la prossima richiesta da servire sarebbe la seconda, perché il suo primo blocco ha indice più vicino a 50.

I parametri di ingresso delle funzionalità sono solo indicativi. Gestire opportunamente le situazioni di errore, tranne gli *overflow* dovuti a valori assoluti troppo elevati e l'inserimento di dati in formato errato da *stdin*.

REGOLE

- Si può utilizzare ogni genere di manuale o di altro materiale didattico
- Per superare la prova, il programma deve contenere l'implementazione delle funzionalità 1, 2 e 3, e deve essere perfettamente funzionante nelle parti 1 e 2. Il voto ottenuto in questo caso è 18.
- Ciascuna funzionalità DEVE essere implementata mediante almeno una funzione.
- Il voto massimo (almeno 30) si ottiene se
 - a) il programma è perfettamente funzionante in ogni sua parte
 - b) tutti i principi di ingegneria del codice visti nel corso sono stati applicati