

8 aprile 2003

La traccia

Si progetti, mediante flow-chart o linguaggio strutturato, una procedura che realizzi, per un memory manager a partizionamento dinamico, l'**algoritmo di aggiornamento della Free Space Table** (che si suppone organizzata secondo la **strategia best-fit**) al termine dell'esecuzione di un programma. Si codifichi quindi la procedura in linguaggio C.

In particolare si vuole che la procedura, ricevuti in input il numero N degli spazi liberi, i loro indirizzi iniziali e finali $INIZIO(i)$ e $FINE(i)$ e gli indirizzi iniziale e finale $INIZIOPROG$ e $FINEPROG$ del programma completato, restituisca aggiornati i due vettori $INIZIO(i)$ e $FINE(i)$.

I requisiti

L'algoritmo di aggiornamento della Free Space Table secondo la strategia best-fit mantiene gli elementi della tabella ordinati in termini di dimensioni crescenti degli spazi liberi. Quando un programma viene completato, la tabella viene perciò aggiornata tenendo conto dello spazio rilasciato dal programma e ordinata nuovamente.

L'analisi

Per rendere agevole l'aggiornamento bisognerà che la Free Space Table sia ordinata in termini di indirizzi iniziali crescenti degli spazi liberi. Ciò significa che il vettore $INIZIO$ e, conseguentemente, il vettore $FINE$ dovranno essere ordinati in tal senso.

Si dovrà verificare a quale degli spazi liberi è adiacente l'area rilasciata dal programma completato e tener conto delle modifiche che sono determinate.

L'area rilasciata dal programma completato può:

1. essere adiacente ad uno spazio libero pre-esistente perchè, per un qualche valore di i risulta $INIZIOPROG = FINE(i) + 1$;
2. essere adiacente ad uno spazio libero pre-esistente perchè, per un qualche valore di i risulta $FINEPROG = INIZIO(i) - 1$;
3. essere adiacente a due spazi liberi pre-esistenti perchè, per un qualche valore di i risulta contemporaneamente $INIZIOPROG = FINE(i) + 1$ e $FINEPROG = INIZIO(i+1) - 1$. In tal caso il numero N degli spazi liberi va decrementato di 1;
4. non essere adiacente ad alcuno degli spazi liberi pre-esistenti; in tal caso si crea un nuovo spazio libero ed il numero N degli spazi liberi va incrementato di 1.

A seconda dei casi bisognerà aggiornare conseguentemente il valore degli opportuni elementi dei vettori $INIZIO$ e $FINE$.

Al termine bisognerà ordinare nuovamente i due vettori $INIZIO$ e $FINE$, questa volta in termini di dimensioni crescenti degli spazi liberi $DIM(i) = FINE(i) - INIZIO(i)$.

Il progetto

Si assume di ordinare in senso crescente i vettori $INIZIO$ e $FINE$ con l'algoritmo di *selection sort*.





