

# **Progetto II - La programmazione C**

Istruzioni e testi degli esercizi  
per il corso di  
Laboratorio di Sistemi Operativi  
A.A. 2008-2009

# Modalità di superamento della seconda parte del corso

Per superare la seconda parte del corso di Laboratorio di Sistemi Operativi, è necessario svolgere, individualmente o in un gruppo di due persone, il progetto relativo alla programmazione C che consiste in un esercizio a scelta tra i due proposti nelle successive slide.

Ogni esercizio richiede la progettazione e l'implementazione di un programma C, opportunamente commentato. Ogni programma dovrà occuparsi della gestione degli errori e produrre output comprensibile.

L'esercizio scelto dovrà essere descritto in una breve relazione in formato OpenDocument che illustri le caratteristiche salienti del programma proposto e fornisca una casistica di test significativi per i diversi possibili input.

Il progetto dovrà essere inviato al docente del corso tramite e-mail entro il 15 luglio 2009. Dopo tale data sarà necessario seguire le istruzioni del corso per l'anno successivo. Il voto, che concorrerà alla formulazione del voto finale dell'esame di Sistemi Operativi, sarà espresso in trentesimi. L'eventuale lode sarà assegnata ai progetti che si saranno contraddistinti.

## Premessa comune agli esercizi: Tabella hash (I)

Una **tabella hash** è una struttura dati che associa chiavi a valori ed è progettata per permettere ricerche in tempo costante.

Mediante una funzione, detta appunto di hash, ogni chiave è trasformata in un indice numerico che identifica una riga della tabella e permette di recuperare rapidamente i valori associati alla chiave data.

Questo comportamento si ottiene comunemente per tabelle di capienza sufficientemente superiore alla quantità di dati inseriti e con funzioni di hash progettate in modo tale da limitare l'eventualità che chiavi distinte vengano associate allo stesso indice.

Poiché questa eventualità si verifica anche in caso di tabelle e funzioni ben progettate, è necessario definire delle strategie per risolvere eventuali conflitti.

## Premessa comune agli esercizi: Tabella hash (II)

La più semplice strategia è la **scansione lineare**:

- per inserire una tupla con chiave e valori viene utilizzata la funzione di hash sulla chiave per determinare l'indice: se l'indice corrisponde ad una riga libera, la tupla viene salvata in quella riga, altrimenti la tupla viene salvata nella prima riga libera successiva;
- per cercare una tupla si usa la funzione di hash sulla chiave e poi si controlla se la chiave data corrisponde alla chiave della riga con l'indice trovato: se è così si restituiscono i valori contenuti in quella riga, altrimenti si passa a controllare le righe successive fino a trovare quella in cui la chiave memorizzata corrisponde alla chiave data.

Ogni esercizio richiede di implementare una tabella hash che segua la strategia illustrata e sia basata su un vettore di strutture contenenti la chiave ed i valori. Si utilizzi  $\text{funzhash(chiave)} = ((\text{ASCII}(\text{car1}) + \text{ASCII}(\text{car2})) \bmod (\text{dimtab}))$ , come funzione di hash, sapendo che per ottenere il valore ASCII di un char basta eseguire il cast ad int e che l'operatore per il modulo è %.

## **Esercizio 1: Rubrica e-mail**

Si progetti e implementi un programma C che legga coppie di dati nome - e-mail dal file passato come parametro e salvi questi dati in una tabella hash.

Il programma dovrà permettere all'utente di specificare un nome sullo standard input. Dopo che l'utente avrà premuto il tasto invio, il programma cercherà l'indirizzo e-mail corrispondente e lo stamperà sullo standard output, segnalando un errore nel caso in cui non sia presente. Il programma dovrà ripetere il comportamento descritto finché l'utente non premerà Ctrl-D.

## Esercizio 2: Traduttore di testi

Si progetti e implementi un programma C che funga da traduttore di testi, basandosi su un vocabolario implementato come una hash table.

Concretamente:

- definire una tabella di lunghezza 22 di coppie parola, traduzione;
- inizializzarla con stringhe vuote;
- definire la funzione hash **funzhash** per riempire ed utilizzare la tabella;
- riempire la tabella con 11 coppie parola, traduzione, leggendo i dati dal file passato come primo parametro;
- applicare **funzhash** per tradurre il file di testo passato come secondo parametro, traducendo una parola con la stringa vuota nel caso non sia presente nel dizionario.