

## Secondo modulo

N.Galesi            S.Guerrini

A.A. 2004/05

Il secondo modulo è dedicato alla realizzazione e all'implementazione delle funzioni e dei dati per l'inserimento nel dizionario degli ipertesti delle informazioni concernenti il contenuto delle pagine web. Nel primo modulo tale dizionario era già stato inizializzato e parzialmente riempito, inserendovi solo l'informazione relativa al nome dei file associati a tutti gli ipertesti contenuti nella directory di lavoro.

Nel secondo modulo ci occuperemo di aggiornare il campo **info** del record **tabelem** di ogni pagina web.

Per ogni pagina web salveremo informazioni relative alle parole contenute nell'ipertesto ed informazioni relative alla struttura dei link tra le differenti pagine web.

### Dizionario locale: strutture dati

Per ogni parola contenuta nel file sorgente di un ipertesto, avremo bisogno di conoscere informazioni relative al suo tipo, alla sua dimensione, al suo stile, alla posizione nel file, se la parola è un link e se lo è, a che ipertesto e la posizione del link nella pagina. Le seguenti dichiarazioni implementano tali informazioni

```
typedef enum{small, normal, large} size;

struct info_word {
    char *parola /* parola */
    int word_pos; /* posizione parola nel file */
    int title; /* vero se la parola nel titolo */
    int italico ; /* vero se la parola in corsivo */
}
```

```

    int bold ; /* vero se la parola in grassetto */
    size word_size /* dimensione */
    char *file_link /* in caso di link attivo, contiene il nome
                        della pagina (file) del link */
    int link_pos /* posizione link nel file */
}

```

Come già descritto nella presentazione del progetto, ad ogni pagina web viene associato un *dizionario locale* di tutte le **differenti parole** contenute nella pagina. Implementeremo tale dizionario mediante una lista concatenata in cui ogni elemento contiene, oltre ad un puntatore al prossimo elemento, anche una *hitlist*, ed il numero di occorrenze della parola nell'ipertesto.

Una hitlist a sua volta è una lista formata da tanti elementi quante sono le occorrenze della parola nel file, ed in cui ogni elemento contiene informazioni relative alle diverse occorrenze della parola stessa.

La definizione della hitlist è data dalla seguente dichiarazione:

```

struct elem_hitlist {
    struct info_word *info_occ_word; /* informazioni sulla particolare
                                      occorrenza della parola */
    struct elem_hitlist *next_elem; /* puntatore al prossimo
                                      elemento della hitlist */
}

```

Il *dizionario locale* è quindi definito dalla seguente dichiarazione.

```

struct elem_diz_loc {
    char *parola; /* entry del dizionario locale */
    struct elem_hitlist *hitlist; /* puntatore alla hitlist
                                   corrispondente a parola*/
    int dim; /* numero di occorrenze di parola
              = dimensione hitlist */
    struct elem_diz_loc *next_word; /* prossima entry
                                      nel dizionario locale */
}

```

Nell'analisi del file html di un ipertesto useremo una funzione, fornita dai docenti, che restituisce tutte le informazioni relative alla prossima parola in un file sorgente, costruendone la struttura `info_word` corrispondente.

```

struct info_word *nextword(FILE *f)

```

## Grafo dei link: strutture dati

Per codificare il grafo dei link (si veda l'introduzione al progetto), ad ogni nodo  $N$  del grafo (cioè ad ogni pagina web) associeremo due liste di puntatori ad elementi del dizionario degli ipertesti. La prima lista contiene quelle pagine a cui  $N$  ha un link (`lista_out`) e la seconda (`lista_in`) contiene le pagine web che hanno un link a  $N$ . Ogni elemento di tali liste è definito dalla seguente dichiarazione:

```
struct info_link {
    struct tabellelem *ptr_pagina /* puntatore all'elemento del
                                   dizionario ipertesti*/
    int molteplicita; /* numero link alla pagina */
    struct info_link *next; /* prossimo elemento */
}
```

Le due liste sono quindi definite dalla dichiarazione:

```
struct info_link *lista_in, *lista_out;
```

È importante notare che le liste dei link devono contenere un solo elemento per ogni differente pagina linkata. Ad esempio, se nella stessa pagina un link è ripetuto più volte, si dovrà solo incrementare la molteplicità dell'elemento già presente nella lista corrispondente a quel link e non crearne uno nuovo. D'altra parte se in un tag corrispondente a un link ad una pagina vi sono più parole (Ad esempio in `<a href = "Descrizione.html"> Cliccare qui </a>`) non si deve incrementare la molteplicità dell'elemento corrispondente alla pagina linkata (Descrizione.html nell'esempio) per ogni parola nel tag. A tal fine sarà utile considerare il valore contenuto nel campo `link_pos` della struttura `info_word` che nel caso di due parole nello stesso tag avrà lo stesso valore.

Infine è possibile che in un ipertesto vi sia un link ad una pagina non contenuta nella directory di lavoro. Tale link non dovrà essere salvato nella lista dei link, anche se la parola contenente il link risulterà tale nella `struct info_word` ad essa associata.

## Informazioni di un ipertesto: strutture dati

Il dizionario locale e il grafo rappresentano le informazioni di un ipertesto che dovremo salvare nel dizionario degli ipertesti (la tabella hash). La struttura che definisce tali informazioni e che dovrà quindi essere salvata nel campo **info** di un elemento del dizionario degli ipertesti e quindi definita da:

```
struct info_ipertesto {
    struct info_link *lista_in; /* lista pagine che puntano */
    struct info_link *lista_out; /* lista pagine puntate */
    struct elem_diz_loc *diz_loc /* dizionario locale */
    int pagerank /* pagerank del documento */ }
```

Si noti che nella precedente struttura viene definito anche un campo *pagerank* in cui verrà salvato il valore del pagerank del documento che però verrà calcolato solo nel III modulo.

## Struttura del modulo

La funzione principale del modulo `crea_diz_ipertesti` deve modificare il dizionario degli ipertesti (la tabella hash creata e parzialmente riempita da `create_node_table` del Modulo I), calcolando il dizionario locale e i nuovi link da aggiungere al grafo degli ipertesti, per ogni pagina web salvata nel dizionario.

Il modulo deve essere composto da un file `nodes.h` (che viene fornito) con le dichiarazioni dei dati usati e da un file `modulo2.c` che implementa `crea_diz_ipertesti`. Si deve completare `modulo2.c` creando la funzione `crea_diz_ipertesti`. La funzione può essere implementata usando altre funzioni, tutte da definire in `modulo2.c`.

I docenti forniranno la funzione `next_word` ed un certo numero file di pagine web scritte nel linguaggio html semplificato su cui testare il programma.