

Esercitazione Make e Linker



università di ferrara

DA SEICENTO ANNI GUARDIAMO AVANTI.

Esercizio 1

- Scrivere un'applicazione in C che consenta di aggiungere coppie (nome, numero di telefono) ad una tabella hash
- Il programma deve leggere una riga da tastiera ed interpretarla

Input

- Se il primo carattere è una 'a', deve aggiungere il nome che segue (una stringa priva di spazi) alla tabella associandolo al numero che segue il nome
- Se il primo carattere è una 'f', deve cercare il nome che segue (una stringa priva di spazi) nella tabella e, se presente, restituire il nome, altrimenti restituire il messaggio "non presente"
- Se il primo carattere non è né una 'a' né una 'f', deve uscire



Esempio

```
rzf@cluster ~/lab $ ./phonebook
```

```
a fabrizio 4836
```

```
a marco 4833
```

```
f fabrizio
```

```
Number 4836
```

```
f marco
```

```
Number 4833
```

```
f giacomo
```

```
Name not found
```

```
e
```

```
rzf@cluster ~/lab $
```



Vincoli di progetto

- Si usi la libreria Glib 2 (parte del GNU GTK <http://www.gtk.org/>)
- Documentazione <http://library.gnome.org/devel/glib/stable/>
- Si usi il tipo di dato GHashTable



Funzioni da utilizzare

```
GHashTable* g_hash_table_new(GHashFunc  
    hash_func, GEqualFunc key_equal_func);  
gboolean g_str_equal(gconstpointer v1,  
    gconstpointer v2);  
guint g_str_hash(gconstpointer v);  
void g_hash_table_insert(GHashTable *hash_table,  
    gpointer key,gpointer value);  
gpointer g_hash_table_lookup(GHashTable *hash_table,  
    gconstpointer key);
```



Altre funzioni utili

```
char * fgets ( char * str, int num, FILE *  
    stream ); // per leggere una riga
```

```
int sscanf ( const char * str, const char *  
    format, ...); // per leggere i comandi dalla  
    riga
```



Istruzioni

- Scrivere un file main.c e un Makefile
- L'eseguibile si deve chiamare phonebook
- Per utilizzare le funzioni di GLib
 - includere glib.h nel file main.c
 - fare linking con la libreria glib-2.0
 - pkg-config --cflags glib-2.0: restituisce i flag per la compilazione
 - pkg-config --libs glib-2.0: restituisce i flag per il linking



Opzioni gcc

Generali:

- o <file> Place the output into <file>
- v verbose, display the programs invoked by the compiler

Compilazione

- c Compile and assemble, but do not link
- fpic Generate position-independent code if possible
- I<dir> Add dir to include search path



Opzioni gcc

Linking:

- L<dir> Add dir to library search path
- l<name> Search for library *libname.a* or *libname.so* in library search path. Es: –
lm looks for *libm.a*
- shared Create a shared library



Esercizio 2

Modificare il programma phonebook in modo da utilizzare una libreria per la gestione dell'archivio

La libreria deve usare un array al posto della hash table

Scrivere il codice della libreria (file store.c)

Compilarla come libreria statica (usando ar per creare un file libstore.a)

Collegarla al main



Opzioni ar

r: replace existing or insert new file(s) into the archive

s: create an archive index

Es:

```
ar rs libc.a atoi.o printf.o
```



Interfaccia: store.h

```
typedef struct
{ char * name;
  int number;
} entry;
typedef struct
{ int count;
  entry* array;
} store;
store * new_store(int size);
void store_insert(store * st, char * name, int number);
int find(store * st, char * name);
```



Esercizio 3

Compilare la libreria come dinamica (file libarch.so)

Collegarla al main



Caricare le librerie dinamiche

```
./phonebook
```

```
./phonebook: error while loading shared  
libraries: libstore.so: cannot open shared  
object file: No such file or directory
```

```
rzf@cluster ~/lab/es3 $ export  
LD_LIBRARY_PATH=.:$LD_LIBRARY_  
PATH
```

