

myTree

Andrea Musolino



Dipartimento di Informatica
Università degli studi Sapienza di Roma

Relazione per l'homework del corso di
Sistemi Operativi II
Canale M-Z

Contents

1	Introduzione	2
2	Struttura del codice	2
2.1	char *options(int argc, char **argv)	2
2.2	void print_tree(char *path, int level)	3

1 Introduzione

L'homework consisteva nel scrivere un programma che avesse lo stesso comportamento del comando `tree`, qui di seguito il link alla pagina web del manuale del comando: [tree\(1\)](#). Oltre il comportamento si doveva permettere di specificare le seguenti opzioni:

`-help, -a, -d, -f, -p, -s, -u, -g, -D, -inodes, -r, -t, -dirsfirst, -L`

2 Struttura del codice

Il codice è stato strutturato nella seguente maniera:

1. **mytree_util.h**: file header in cui sono dichiarate le strutture dati ed i prototipi di funzione
2. **mytree_util.c**: file in cui vengono implementate le funzioni dichiarate nel file `.h`
3. **main.c**: file main in cui viene eseguito il programma

La maggior parte del lavoro viene svolto da tali funzioni:

- `char *options(int argc, char **argv)`
- `void print_tree(char *path, int level)`

2.1 `char *options(int argc, char **argv)`

- `int argc`: è il numero di elementi che sono presenti in `argv`
- `char **argv`: contiene tutti gli argomenti che sono stati passati da riga di comando al main

La funzione seleziona le opzioni scelte (verranno impostate delle flag) e restituisce il path inserito dall'utente. Tale compito viene svolto dalla funzione: `getopt_long` ([getopt_long\(3\)](#)). La funzione si comporta allo stesso modo di `getopt` ([getopt\(3\)](#)), ovvero, traduce la lista di argomenti contenuti in `argv` e restituisce il prossimo carattere **conosciuto**. Un carattere è **conosciuto** se è presente nella stringa `optstring`, passata come argomento a `getopt`. A differenza di `getopt` in cui si può specificare solo *caratteri*, con `getopt_long` si possono specificare anche *parole*, tramite un array di tipo `struct option`; struttura i cui campi sono:

- `char *name`: il nome dell'opzione
- `int has_arg`: se l'opzione richiede argomenti
- `int *flag`
- `int val`: corrisponde al opzione corta associata all'opzione

Il path verrà restituito tramite l'intero `optind`, incluso nella libreria `getopt.h`, in cui vi è memorizzato l'indice dell'ultimo carattere **non conosciuto** presente in `argv`.

2.2 void print_tree(char *path, int level)

- `char *path`: il path di una directory
- `int level`: il livello di indentazione

La funzione è la funzione principale del programma. Il flusso è il seguente:

1. Viene aperto la directory tramite la funzione `opendir`
2. Tramite `readdir` si conta il numero di entry all'interno di `path`, saltando la directory corrente e precedente
3. Alloca un array di tipo `struct dirent` con tante locazioni, quante sono le entry contate
4. Riavvolge il puntatore alla directory tramite `rewinddir`, facendo ripartire lo stream dalla prima entry
5. Salva all'interno dell'array le entry
6. Ordina l'array tramite `qsort` alfabeticamente, o diversamente a seconda delle opzioni selezionate
7. Stampa le entry salvate. Se una entry è una directory, si chiama ricorsivamente la funzione incrementando il livello di indentazione di uno.