

# Laboratorio di Elementi di Architetture e Sistemi Operativi

Soluzioni del Compitino del 14 Maggio 2013

## Esercizio 1.

1. Considerare il codice scritto per l'esercizio 2 della lezione scorsa.
2. Creare un Makefile per compilare il codice con il seguente contenuto:

```
main: main.o liste.o
    gcc main.o liste.o -o main
main.o: main.c liste.h
    gcc -c main.c
liste.o: liste.c liste.h
    gcc -c liste.c
clean:
    rm *.o main
```

3. Qual'è il risultato dell'esecuzione del comando `make`?

```
$ make
gcc -c main.c
gcc -c liste.c
gcc main.o liste.o -o main
```

Il comando compila i file sorgente e genera l'eseguibile `main`.

4. Eseguire il comando `touch liste.c`. Cosa succede quando si esegue il comando `make`?

```
$ touch liste.c
$ make
gcc -c liste.c
gcc main.o liste.o -o main
```

In questo caso viene ricompilato solo il file `liste.c` ed eseguito nuovamente il linking per generare l'eseguibile.

5. Eseguire il comando `touch liste.h`. Cosa succede se si esegue il comando `make`? Quali sono le differenze rispetto al punto precedente?

```
$ touch liste.h
$ make
gcc -c main.c
gcc -c liste.c
gcc main.o liste.o -o main
```

In questo caso, poiché anche `main.o` dipende da `liste.h`, vengono ricompilati i file `liste.c` e `main.c`, oltre ad effettuare nuovamente il linking per generare l'eseguibile.

6. Eseguire il comando `make clean` e descriverne il risultato.

Il comando cancella i file oggetto `.o` e l'eseguibile del programma.

7. Modificare il Makefile sostituendo le regole per `main.o` e `liste.o` con la regola implicita

```
%.o : %.c
    gcc -o $@ -c $<
```

8. Eseguire nuovamente `make` e verificarne il funzionamento.

```
$ make
gcc -o main.o -c main.c
gcc -o liste.o -c liste.c
gcc main.o liste.o -o main
```

Il comando `make` ha lo stesso effetto del punto 3 e genera correttamente l'eseguibile.

9. Eseguire il comando `touch liste.h`. Cosa succede se si esegue il comando `make`? Il comportamento è lo stesso Makefile precedente?

```
$ touch liste.h
$ make
make: `main' is up to date.
```

In questo caso il make afferma che non c'è nulla da fare perché main è aggiornato. Questo non è corretto: la modifica di un file header deve provocare la ricompilazione di tutti i file che lo includono.

10. *Modificare il Makefile in modo che le modifiche agli header file causino la corretta ricompilazione dei file oggetto.*

In questo caso è necessario includere esplicitamente le dipendenze dai file header, per esempio aggiungendo le seguenti regole alla fine del Makefile:

```
main.o: liste.h
liste.o: liste.h
```

**Esercizio 2.** *Modificare il Makefile dell'esercizio precedente in modo che esegua le seguenti operazioni:*

1. *generare la libreria statica libliste.a per la gestione delle liste;*
2. *generare l'eseguibile del programma, linkato staticamente alla libreria;*
3. *fornire un target install che esegua le seguenti operazioni:*
  - *copiare l'eseguibile in \$HOME/bin;*
  - *copiare la libreria statica in \$HOME/lib;*
  - *copiare gli header file della libreria in \$HOME/include;*
4. *fornire un target archive che crei un archivio liste.zip usando il comando zip che contenga solamente il Makefile ed i file .c ed .h con il codice;*
5. *fornire un target clean per rimuovere i file oggetto, l'eseguibile, le librerie e l'archivio.*

```
main: main.o libliste.a
gcc -o main main.o -L. -lliste

libliste.a: liste.o
ar r libliste.a liste.o

%.o : %.c
gcc -o $@ -c $<

clean:
rm -f *.o *.a main liste.zip

install: main libliste.a liste.h
cp main $(HOME)/bin
cp libliste.a $(HOME)/lib
cp liste.h $(HOME)/include

archive: liste.zip

liste.zip: *.c liste.h Makefile
zip liste.zip $^

liste.o: liste.h
main.o: liste.h
```

*Verificare il corretto funzionamento del Makefile:*

- *verificare che generi correttamente l'eseguibile;*

```
$ make
gcc -o main.o -c main.c
gcc -o liste.o -c liste.c
ar r libliste.a liste.o
ar: creating archive libliste.a
gcc -o main main.o -L. -lliste
```

- *verificare che i target clean e install eseguano il loro compito correttamente;*

```
$ make clean
rm -f *.o *.a main liste.zip

$ make install
gcc -o main.o -c main.c
gcc -o liste.o -c liste.c
ar r libliste.a liste.o
ar: creating archive libliste.a
gcc -o main main.o -L. -lliste
cp main /Users/davide/bin
cp libliste.a /Users/davide/lib
cp liste.h /Users/davide/include
```

- *verificare che l'archivio creato con make archive non contenga file oggetto, librerie od eseguibili;*

```
$ make archive
zip liste.zip main.c liste.c liste.h Makefile
  adding: main.c (deflated 57%)
  adding: liste.c (deflated 70%)
  adding: liste.h (deflated 58%)
  adding: Makefile (deflated 52%)
$ unzip -l liste.zip
Archive:  liste.zip
  Length      Date    Time    Name
-----
      862   05-28-2013 15:08   main.c
     1621   05-28-2013 15:09   liste.c
      433   05-28-2013 15:15   liste.h
      411   05-28-2013 15:23   Makefile
-----
     3327
                   4 files
```

- *dopo aver estratto l'archivio in una directory separata, verificare se il comando make, eseguito in quella directory, genera correttamente il programma.*

```
$ mkdir prova
$ cd prova
$ unzip ../liste.zip
Archive:  ../liste.zip
  inflating: main.c
  inflating: liste.c
  inflating: liste.h
  inflating: Makefile
$ make
gcc -o main.o -c main.c
gcc -o liste.o -c liste.c
ar r libliste.a liste.o
ar: creating archive libliste.a
gcc -o main main.o -L. -lliste
```