

Cognome: \_\_\_\_\_ Nome: \_\_\_\_\_ Matricola: \_\_\_\_\_ Calcolatore: \_\_\_\_\_

Sistemi Operativi — A.A. 2006-2007, prova pratica del 10 settembre 2007

# Compito A

Vietato comunicare con chiunque. Vietato l'uso di rete, cellulari, floppy disk, pen drive e affini. Libri chiusi. Si può usare tutta la documentazione disponibile sul calcolatore. Non spegnere mai il calcolatore. Se hai problemi con il calcolatore rivolgiti subito al docente. Tempo a disposizione: 60 minuti.

## Leggere attentamente prima di iniziare

- Scarica dall'url <http://192.168.161.70/compito.tar.gz> il pacchetto dei file che ti servono per il compito e scompattalo all'interno della tua home directory. (suggerimento: “wget <http://192.168.161.70/compito.tar.gz> ; tar xvzf compito.tar.gz ”)
- Scrivi nome, cognome, matricola e numero del calcolatore su questo foglio.
- **Scrivi gli stessi dati nel file dati\_studente.txt.**
- Prepara un documento di identità a portata di mano.
- **Metti tutti i file che crei durante il compito nella directory ~/compito dentro la home** (nel seguito la tilde “~” significa “home dell'utente”).
- Gli esercizi sono etichettati con 1,2 o 3 asterischi:
  - \* = esercizio semplice, valutazione alta, risolvi questi prima degli altri
  - \*\* = esercizio di media difficoltà
  - \*\*\* = esercizio difficile, valutazione bassa, risolvi dopo aver risolto gli altri

## Istruzioni per la consegna del compito

Non spegnere il calcolatore e recati dal docente con questo foglio compilato.

## Esercizio 1

Il file di testo packages.txt contiene un record per ciascun pacchetto software della distribuzione linux debian. Nel file ciascun record è separato da una linea vuota, i campi sono su linee distinte, tranne il campo "Description" che è su più linee. Alcuni campi possono mancare. Per svolgere l'esercizio non è necessario conoscere il significato di tutti i campi. (Suggerimento: per il processamento di questo file tramite awk considera la possibilità di porre RS="", cioè stringa vuota, e FS="\n", considera la funzione gsub come strumento di cancellazione).

1. \* Il campo “Depends” contiene una espressione che indica le dipendenze da altri pacchetti. Ciascuna dipendenza può avere dei vincoli sulla versione. Ad esempio “libm (>= 3.5.6-3)” significa che il pacchetto ha bisogno di libm con una versione superiore o uguale a quella indicata. Contare i pacchetti che hanno bisogno di libc6 con una versione superiore o uguale a 2.4.

Scrivi nel file ~/compito/esercizio1/soluzione1.1.txt il comando usato e il suo output (fai copia-e-incolla dal terminale).

2. \*\* Nelle espressioni del campo “Depends” compaiono nomi di package. Dai un comando che mostri una tabella in cui la prima colonna elenchi i packages che compaiono nei campi Depends e la seconda il numero di volte che quel package compare. Per rispondere all'adomanda devi “ripulire” l'input da tutto cio' che non e' un nome di package.

Scrivi nel file ~/compito/esercizio1/soluzione1.2.txt il comando usato e il suo output (fai copia-e-incolla dal terminale).

3. \*\*\* [questo punto richiede più tempo degli altri, svolgilo per ultimo] Scrivere uno script che elabori i dati da packages.txt e produca su standard output il seguente report:

- i record sono separati da una riga vuota
- esiste un record per ciascun valore del campo “Section”
- le restanti righe del record contengono i nomi dei pacchetti (campo “Package”) che hanno quel valore di “Section” con a fianco il valore di “Version”
- i record devono essere ordinati in modo che le “Section” con più pacchetti appaiano in testa.

Il nome del tuo script deve essere ~/compito/esercizio1/script.sh

## Esercizio 2

Il programma "fib" contenuto in prj1 è composto da più file con estensione ".c" e ".h". Tale programma crea una list contenete i logaritmi naturali dei primi n numeri di fibonacci e ne stampa il contenuto (n e' passato come parametro). La directory prj2 contiene una copia di prj1 che devi modificare per rispondere alle seguenti domande.

1. \* Usa il comando "gcc \*.c" per una prima compilazione. Correggi gli errori di preprocessamento, di compilazione e di link che incontri. Mostra tutti gli errori e spiegali man mano che li correggi. Mostra quindi l'output di compilazione ed esecuzione del programma corretto. Nota che per eseguire il programma devi fornire un parametro. Usa, ad esempio, "6".

Per correggere gli errori modifica la copia in ~/compito/esercizio2/prj2 e lascia intatto ~/compito/esercizio2/prj1.

Scrivi in ~/compito/esercizio2/soluzione2.1.txt la soluzione di questo esercizio (fai copia-e-incolla dal terminale per gli errori e l'output).

2. \*\* Crea un Makefile con i seguenti target **badando a mettere le dipendenze opportune**
  - fib\_static: crea l'eseguibile "fib\_static" linkato staticamente senza simboli di debug
  - fib\_dynamic: crea l'eseguibile "fib\_dynamic" linkato dinamicamente con simboli di debug
  - clean: pulisce il progetto cancellando i file inutili (\*.o, \*~)
  - delete: come clean ma cancella anche i target
  - test: lancia fib\_dynamic con parametro "6".

Il Makefile si deve chiamare ~/compito/esercizio2/prj2/Makefile

3. \* Crea una patch tra prj1 e prj2 e provala su una copia "prj1copy" che crei tu.

Metti nel file ~/compito/esercizio2/soluzione2.3.txt i comandi per creare e per applicare la patch (fai copia-e-incolla dal terminale) . Metti la patch nel file ~/compito/esercizio2/soluzione2.3\_patch.txt.

4. \*\* Mostra il disassemblato della funzione fib contenuto nel file init\_list.o.

Metti nel file ~/compito/esercizio2/soluzione2.4.txt i comandi per portare a termine l'esperimento (fai copia-e-incolla dal terminale) e la risposta alla domanda.

5. \*\* Considera una esecuzione di fib\_dynamic con parametro 16. Considera la 2010-ima esecuzione della funzione fib(), mostra il contenuto della variabile "f" subito prima del return di tale esecuzione.

Scrivi nel file ~/compito/esercizio2/soluzione2.5.txt la sessione di debug (fai copia-e-incolla dal terminale).