

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

## Sistemi Operativi — A.A. 2005-2006, prova pratica del 21 settembre 2006

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

### 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. 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")

1. \* Il campo “Maintainer” contiene una stringa con l'email della persona che mantiene il pacchetto. Mostra un comando che dia il numero di maintainer con email nel top domain “.de” o “.org”.  
Scrivi nel file ~/compito/esercizio1/soluzione1.1.txt il comando usato e il suo output (fai copia-e-incolla dal terminale).
2. \*\*Il campo “Package” (sempre il primo) contiene il nome del pacchetto, il campo “Installed-Size” (sempre il quarto) i KB occupati dopo l'installazione e il campo “Priority” (sempre il secondo) l'importanza del pacchetto. Dai un comando che mostri una tabella con una riga per ciascun pacchetto con priorità “optional” con due colonne: “nome del pacchetto”, “KB dopo installazione”.  
La tabella deve essere ordinata in ordine decrescente dei KB.  
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 maintainer , il suo nome ed indirizzo di email è indicato nella prima riga del record (suggerimento: sostituisci gli spazi con un “\_”)
  - le restanti righe del record contengono i nomi dei pacchetti mantenuti
  - i record devono essere ordinati per quantità di pacchetti mantenuti (prima i maintainer con molti pacchetti)

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

## Esercizio 2

Il programma contenuto in prj1 è composto da più file .c. Tale programma crea una lista contenete dei valori numerici, ne stampa il contenuto in ordine inverso ed esce. 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. Il progetto non si compila e non si linka. Mostra **tutti** gli errori, di compilazione e/o link e spiegali man mano che li correggi. Nel correggere porta **tutte** le strutture relative alla lista nel file lista.h. Mostra quindi l'output di compilazione ed esecuzione del programma corretto.

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

2. \* Crea una patch tra prj1 e prj2 e provane il funzionamento su una copia di prj1 che devi fare tu. Metti nel file ~/compito/esercizio2/soluzione2.2.txt i comandi per creare e per applicare la patch. Metti la patch nel file ~/compito/esercizio2/soluzione2.2\_patch.txt.

3. \*\* Crea un Makefile con i seguenti target badando a mettere le dipendenze opportune
  - main: crea l'eseguibile linkato dinamicamente senza simboli di debug
  - main\_debug: crea eseguibile linkato staticamente con simboli di debug
  - clean: pulisce il progetto cancellando i file inutili (\*.o, \*.~)
  - delete: come clean ma cancella anche i target
  - inodes: mostra in output il contenuto degli inodes dei file main e main\_debug

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

4. \*\* Considera la funzione crea(). Mostra l'esecuzione step-by-step della prima iterazione. Mostra l'esecuzione di 10 istruzioni dopo che  $x$  ha assunto un valore maggiore di 261400. Mostra il disassemblato della funzione crea.

Scrivi nel file ~/compito/esercizio2/soluzione2.4.txt la sessione di debug (fai copia-e-incolla dal terminale).

## Istruzioni per la consegna del compito

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