Cognome:_____ Nome: _____ Matricola:_____ Calcolatore: _____

Sistemi Operativi 1 — A.A. 2004-2005, prova pratica del 20 settembre 2005

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 "~" signifca "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 "Architecture" contiene un valore che identifica l'architettura su cui il pacchetto software funziona. Mostra un comando che dia il numero di pacchetti che funzionano sulla architettura all.

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

2. **Il campo "Section" (sempre il terzo del record) contiene una stringa che identifica la sezione tematica in cui il pacchetto viene classificato. Il campo "Installed-Size:" (sempre il quarto del record) contiene il numero di kilobytes installati. Mostra un comando che calcoli la somma dei valori di Installed-Size nella sezione "libs".

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 legga packages.txt e produca su standard output frasi del tipo "X server per Y" dove X e Y sono nomi di pacchetti. In ciascun record, il nome del pacchetto è il valore del campo "Package", le dipendenze tra pacchetti sono espresse nel campo "Depends". Il campo "Depends" ha una sintassi complessa, ignora le virogle e le pipes "|" e tutto cio' che sta tra parentesi, come nel seguente esempio:

```
Package: aspell
Depends: aspell-bin (>1.3), aspell-en(<=3) | aspell6-dictionary
```

deve generare...

```
aspell-bin serve per aspell
aspell-en serve per aspell
aspell6-dictionary serve per aspell
```

Suggerimento: processa il file con awk riga per riga e non record per record.

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

Esercizio 2

Il programma contenuto in prj1 è composto da più file .c e crea una lista contenete dei valori numerici, ne stampa il contenuto in oridne inverso ed esce. La directory prj2 contiene una copia di prj1 che devi modificare per rispondere alle seguenti domande.

1. * Il progetto è dotato di makefile per la compilazione, usalo per compilare. Noterai che 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.

Per correggere gli errori modifca la copia in ~/compito/esercizio2/prj2 e lascia intatto ~/compito/esercizio2/prj1.
Metti i comandi usati, i vari errori (l'output dei comandi) e la tua interpretazione in ~/compito/esercizio2/soluzione2.1.txt

2. *Il programma corretto.

Mostra l'output dell'esecuzione programma corretto in ~/compito/esercizio2/soluzione2.2.txt

3. *Crea una patch tra prj1 e prj2

Metti la patch nel file ~/compito/esercizio2/soluzione2.3_patch.txt

4. **Aggiungi a ~/compito/esercizio2/prj2/Makefile i seguenti target
   - clean: pulisce il progetto cancellando i file inutili (*.o, *.~)
   - delete: come clean ma cancella anche i target
   - crealista_debug: crea eseguibile linkato dinamicamente con simboli di debug
   - main.s: deve contenere il codice assembly derivato da main.c (consulta il manuale del gcc per l'opzione giusta)

Modifica il file ~/compito/esercizio2/prj2/Makefile

5. ** Considera la lista dopo l'esecuzione della funzione crea(), mostra il contenuto di ciascun campo dei primi due elementi subito dopo la creazione della lista e prima delle modifiche. Mostra anche l'indirizzo di memoria della variabile X di main.

Scrivi nel file ~/compito/esercizio2/soluzione2.5.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.

Cognome:_____ Nome: _____ Matricola:_____ Calcolatore: _____

Sistemi Operativi 1 — A.A. 2004-2005, prova pratica del 20 settembre 2005

Compito B

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 compito/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 "~" signifca "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 "Suggests" (campo opzionale) contiene un valore che descrive i pacchetti suggeriti da usare assieme a quello descritto dal record (la sintassi usata in questo campo non ti interessa). Mostra un comando che dia il numero di pacchetti che nel campo "Suggests" contegono la stringa "apache".

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

2. **Il campo "Maintainer" (sempre il quinto del record) contiene una stringa che identifica la sezione tematica in cui il pacchetto viene classificato. Il campo "Installed-Size:" (sempre il quarto del record) contiene il numero di kilobytes installati. Mostra un comando che calcoli la somma dei valori di Installed-Size per il mainteiner "Gustavo Noronha Silva".

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 legga packages.txt e produca su standard output frasi del tipo "X server per Y" dove X e Y sono nomi di pacchetti. In ciascun record, il nome del pacchetto è il valore del campo "Package", le dipendenze tra pacchetti sono espresse nel campo "Depends". Il campo "Depends" ha una sintassi complessa, ignora le virogle e le pipes "|" e tutto cio' che sta tra parentesi, come nel seguente esempio:

```
Package: aspell
Depends: aspell-bin (>1.3), aspell-en(<=3) | aspell6-dictionary
```

deve generare...

```
aspell-bin serve per aspell
aspell-en serve per aspell
aspell6-dictionary serve per aspell
```

Suggerimento: processa il file con awk riga per riga e non record per record.

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

Esercizio 2

Il programma contenuto in prj1 è composto da più file .c e crea una lista contenete dei valori numerici, ne stampa il contenuto in oridne inverso ed esce. La directory prj2 contiene una copia di prj1 che devi modificare per rispondere alle seguenti domande.

1. * Il progetto è dotato di makefile per la compilazione, usalo per compilare. Noterai che 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.

Per correggere gli errori modifca la copia in ~/compito/esercizio2/prj2 e lascia intatto ~/compito/esercizio2/prj1.
Metti i comandi usati, i vari errori (l'output dei comandi) e la tua interpretazione in ~/compito/esercizio2/soluzione2.1.txt

2. *Il programma corretto.

Mostra l'output dell'esecuzione programma corretto in ~/compito/esercizio2/soluzione2.2.txt

3. *Crea una patch tra prj1 e prj2

Metti la patch nel file ~/compito/esercizio2/soluzione2.3_patch.txt

4. **Aggiungi a ~/compito/esercizio2/prj2/Makefile i seguenti target
   - clean: pulisce il progetto cancellando i file inutili (*.o, *.~)
   - delete: come clean ma cancella anche i target
   - crealista_debug: crea eseguibile linkato dinamicamente con simboli di debug
   - main.i: deve contenere il codice di main.c dopo la fase di preprocessamento (consulta il manuale del gcc per l'opzione giusta)

Modifica il file ~/compito/esercizio2/prj2/Makefile

5. ** Considera la lista dopo l'esecuzione della funzione crea(), mostra il contenuto di ciascun campo degli ultimi due elementi subito dopo la creazione della lista e prima delle modifiche. Mostra anche l'indirizzo di memoria della funzione stampa().

Scrivi nel file ~/compito/esercizio2/soluzione2.5.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.