

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

**Sistemi Operativi — A.A. 2005-2006, prova pratica del 28 aprile 2006**

# 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

## Esercizio 1

Il file di testo `radb.db.txt` contiene un record per ciascun prefisso IP registrato presso un certo registro di Internet. Nel file ciascun record è separato da una linea vuota, i campi sono su linee distinte, tranne il campo “`descr`” che è su più linee. Per svolgere l'esercizio non è necessario conoscere il significato di tutti i campi. (Suggerimento: il processamento di tale file con `awk` può essere fatto riga per riga oppure record per record ponendo `RS=""`, cioè stringa vuota, e `FS="\n"`)

1. \* Il campo “`route`” (sempre il primo) contiene il prefisso. Mostra un comando che dia il numero di prefissi che terminano per `/24`.

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

2. \*\* Il campo “`origin`” contiene una stringa che identifica l'autonomous system (AS) che origina il prefisso. Considera gli AS che originano almeno un prefisso che inizi per `192`. Mostra un comando che calcoli una tabella con due colonne: una colonna contiene gli AS che originano almeno un prefisso che inizi per `192`, l'altra, per ciascun AS il numero di tali prefissi originati.

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 `radb.db.txt` e produca su standard output il seguente report:

- i record sono separati da una riga vuota
- esiste un record per ciascun AS originator, tale AS è indicato nella prima riga del record
- le restanti righe del record contengono tutti i prefissi originati da quell'AS
- i record devono essere ordinati per numero crescente di AS.

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 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. 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. Nel correggere porta **tutte** le strutture relative alla lista in un file list.h.

Per correggere gli errori modifica 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 e provane il funzionamento su una copia di prj1 che devi fare tu.

Metti nel file ~/compito/esercizio2/soluzione2.3.txt i comandi per creare e per applicare la patch. Metti la patch nel file ~/compito/esercizio2/soluzione2.3\_patch.txt.

4. \*\* Crea un Makefile con i seguenti target

- main: crea l'eseguibile linkato staticamente senza simboli di debug

- clean: pulisce il progetto cancellando i file inutili (\*.o, \*.~)

- delete: come clean ma cancella anche i target

- main\_debug: crea eseguibile linkato dinamicamente con simboli di debug

- result.txt: mette l'output dell'esecuzione in result.txt

- test: esegue main e fa un diff tra result.txt e l'output dell'esecuzione mostrando il risultato su standard output (puoi usare un file temporaneo se vuoi).

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

5. \*\* Considera la funzione createList(). Mostra il valore della variabile  $x$  all'inizio della seconda iterazione. Mostra il valore della variabile  $x$  all'inizio della novantesima iterazione.

Considera la lista dopo la creazione. Mostra il contenuto di ciascun campo del penultimo elemento della lista. Mostra anche l'indirizzo di memoria in cui è memorizzata la variabile theList.

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 — A.A. 2005-2006, prova pratica del 28 aprile 2006

## 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 “~” 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 radb.db.txt contiene un record per ciascun prefisso IP registrato presso un certo registro di Internet. Nel file ciascun record è separato da una linea vuota, i campi sono su linee distinte, tranne il campo "descr" che è su più linee. Per svolgere l'esercizio non è necessario conoscere il significato di tutti i campi. (Suggerimento: il processamento di tale file con awk può essere fatto riga per riga oppure record per record ponendo RS="", cioè stringa vuota, e FS="\n")

1. \* Il campo “route” (sempre il primo) contiene il prefisso. Mostra un comando che dia il numero di prefissi che iniziano per 192.  
Scrivi nel file ~/compito/esercizio1/soluzione1.1.txt il comando usato e il suo output (fai copia-e-incolla dal terminale).
2. \*\* Il campo “origin” contiene una stringa che identifica l'autonomous system (AS) che origina il prefisso. Considera gli AS che originano almeno un prefisso che finisca per /24. Mostra un comando che calcoli una tabella con due colonne: una colonna contiene gli AS che originano almeno un prefisso che finisca per /24, l'altra, per ciascun AS il numero di tali prefissi originati.  
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 radb.db.txt e produca su standard output il seguente report:
  - i record sono separati da una riga vuota
  - esiste un record per ciascun AS originator, tale AS è indicato nella prima riga del record
  - le restanti righe del record contengono tutti i prefissi originati da quell'AS
  - i record devono essere ordinati per numero crescente di AS.

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 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. 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. Nel correggere porta **tutte** le strutture relative alla lista in un file list.h.

Per correggere gli errori modifica 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 e provane il funzionamento su una copia di prj1 che devi fare tu.

Metti nel file ~/compito/esercizio2/soluzione2.3.txt i comandi per creare e per applicare la patch. Metti la patch nel file ~/compito/esercizio2/soluzione2.3\_patch.txt.

4. \*\* Crea un Makefile con i seguenti target

- main: crea l'eseguibile linkato staticamente senza simboli di debug

- clean: pulisce il progetto cancellando i file inutili (\*.o, \*.~)

- delete: come clean ma cancella anche i target

- main\_debug: crea eseguibile linkato dinamicamente con simboli di debug

- result.txt: mette l'output dell'esecuzione in result.txt

- test: esegue main e fa un diff tra result.txt e l'output dell'esecuzione mostrando il risultato su standard output (puoi usare un file temporaneo se vuoi).

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

5. \*\* Considera la funzione createList(). Mostra il valore della variabile  $x$  all'inizio della seconda iterazione. Mostra il valore della variabile  $x$  all'inizio della novantesima iterazione.

Considera la lista dopo la creazione. Mostra il contenuto di ciascun campo del penultimo elemento della lista. Mostra anche l'indirizzo di memoria in cui è memorizzata la variabile theList.

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 — A.A. 2005-2006, prova pratica del 28 aprile 2006**

## Compito C

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 `radb.db.txt` contiene un record per ciascun prefisso IP registrato presso un certo registro di Internet. Nel file ciascun record è separato da una linea vuota, i campi sono su linee distinte, tranne il campo “`descr`” che è su più linee. Per svolgere l'esercizio non è necessario conoscere il significato di tutti i campi. (Suggerimento: il processamento di tale file con `awk` può essere fatto riga per riga oppure record per record ponendo `RS=""`, cioè stringa vuota, e `FS="\n"`)

1. \* Il campo “`changed`” contiene un indirizzo di email. Mostra un comando che dia il numero di record in cui questo indirizzo di email finisce per “.net”.  
Scrivi nel file `~/compito/esercizio1/soluzione1.1.txt` il comando usato e il suo output (fai copia-e-incolla dal terminale).
2. \*\* Il campo “`origin`” contiene una stringa che identifica l'autonomous system (AS) che origina il prefisso. Considera gli AS che originano almeno un prefisso che inizi per 192. Mostra un comando che calcoli una tabella con due colonne: una colonna contiene gli AS che originano almeno un prefisso che inizi per 192, l'altra, per ciascun AS il numero di tali prefissi originati.  
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 `radb.db.txt` e produca su standard output il seguente report:
  - i record sono separati da una riga vuota
  - esiste un record per ciascun AS originator, tale AS è indicato nella prima riga del record
  - le restanti righe del record contengono tutti i prefissi originati da quell'AS
  - i record devono essere ordinati per numero crescente di AS.

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 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. 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. Nel correggere porta **tutte** le strutture relative alla lista in un file list.h.

Per correggere gli errori modifica 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 e provane il funzionamento su una copia di prj1 che devi fare tu.

Metti nel file ~/compito/esercizio2/soluzione2.3.txt i comandi per creare e per applicare la patch. Metti la patch nel file ~/compito/esercizio2/soluzione2.3\_patch.txt.

4. \*\* Crea un Makefile con i seguenti target

- main: crea l'eseguibile linkato staticamente senza simboli di debug

- clean: pulisce il progetto cancellando i file inutili (\*.o, \*.~)

- delete: come clean ma cancella anche i target

- main\_debug: crea eseguibile linkato dinamicamente con simboli di debug

- result.txt: mette l'output dell'esecuzione in result.txt

- test: esegue main e fa un diff tra result.txt e l'output dell'esecuzione mostrando il risultato su standard output (puoi usare un file temporaneo se vuoi).

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

5. \*\* Considera la funzione createList(). Mostra il valore della variabile  $x$  all'inizio della seconda iterazione. Mostra il valore della variabile  $x$  all'inizio della novantesima iterazione.

Considera la lista dopo la creazione. Mostra il contenuto di ciascun campo del penultimo elemento della lista. Mostra anche l'indirizzo di memoria in cui è memorizzata la variabile theList.

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 — A.A. 2005-2006, prova pratica del 28 aprile 2006

## Compito D

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 “~” 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 radb.db.txt contiene un record per ciascun prefisso IP registrato presso un certo registro di Internet. Nel file ciascun record è separato da una linea vuota, i campi sono su linee distinte, tranne il campo "descr" che è su più linee. Per svolgere l'esercizio non è necessario conoscere il significato di tutti i campi. (Suggerimento: il processamento di tale file con awk può essere fatto riga per riga oppure record per record ponendo RS="", cioè stringa vuota, e FS="\n")

1. \* Il campo “origin” contiene l'AS che origina il prefisso. Mostra un comando che dia il numero di prefissi originati da AS701.  
Scrivi nel file ~/compito/esercizio1/soluzione1.1.txt il comando usato e il suo output (fai copia-e-incolla dal terminale).
2. \*\* Il campo “origin” contiene una stringa che identifica l'autonomous system (AS) che origina il prefisso. Considera gli AS che originano almeno un prefisso che finisca per /24. Mostra un comando che calcoli una tabella con due colonne: una colonna contiene gli AS che originano almeno un prefisso che finisca per /24, l'altra, per ciascun AS il numero di tali prefissi originati.  
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 radb.db.txt e produca su standard output il seguente report:
  - i record sono separati da una riga vuota
  - esiste un record per ciascun AS originator, tale AS è indicato nella prima riga del record
  - le restanti righe del record contengono tutti i prefissi originati da quell'AS
  - i record devono essere ordinati per numero crescente di AS.

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 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. 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. Nel correggere porta **tutte** le strutture relative alla lista in un file list.h.

Per correggere gli errori modifica 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 e provane il funzionamento su una copia di prj1 che devi fare tu.

Metti nel file ~/compito/esercizio2/soluzione2.3.txt i comandi per creare e per applicare la patch. Metti la patch nel file ~/compito/esercizio2/soluzione2.3\_patch.txt.

4. \*\* Crea un Makefile con i seguenti target

- main: crea l'eseguibile linkato staticamente senza simboli di debug

- clean: pulisce il progetto cancellando i file inutili (\*.o, \*.~)

- delete: come clean ma cancella anche i target

- main\_debug: crea eseguibile linkato dinamicamente con simboli di debug

- result.txt: mette l'output dell'esecuzione in result.txt

- test: esegue main e fa un diff tra result.txt e l'output dell'esecuzione mostrando il risultato su standard output (puoi usare un file temporaneo se vuoi).

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

5. \*\* Considera la funzione createList(). Mostra il valore della variabile  $x$  all'inizio della seconda iterazione. Mostra il valore della variabile  $x$  all'inizio della novantesima iterazione.

Considera la lista dopo la creazione. Mostra il contenuto di ciascun campo del penultimo elemento della lista. Mostra anche l'indirizzo di memoria in cui è memorizzata la variabile theList.

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 — A.A. 2005-2006, prova pratica del 28 aprile 2006

## Compito E

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 radb.db.txt contiene un record per ciascun prefisso IP registrato presso un certo registro di Internet. Nel file ciascun record è separato da una linea vuota, i campi sono su linee distinte, tranne il campo "descr" che è su più linee. Per svolgere l'esercizio non è necessario conoscere il significato di tutti i campi. (Suggerimento: il processamento di tale file con awk può essere fatto riga per riga oppure record per record ponendo RS="", cioè stringa vuota, e FS="\n")

1. \* Il campo “changed” riporta alla fine della riga la data dell'ultimo cambiamento. Mostra un comando che dia il numero di prefissi che sono stati cambiati nel mese di marzo (marzo=03), giorno e anno qualsiasi.

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

2. \*\* Il campo “origin” (sempre presente prima di “changed”) contiene una stringa che identifica l'autonomous system (AS) che origina il prefisso. Considera gli AS origin per i record cambiati nel primo giorno del mese. Mostra un comando che calcoli una tabella con due colonne: una colonna contiene gli AS che originano almeno un prefisso di un record cambiato il primo giorno del mese (=01), mese e anno qualsiasi, l'altra, per ciascun AS il numero di tali prefissi originati.

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 radb.db.txt e produca su standard output il seguente report:

- i record sono separati da una riga vuota
- esiste un record per ciascun AS originator, tale AS è indicato nella prima riga del record
- le restanti righe del record contengono tutti i prefissi originati da quell'AS
- i record devono essere ordinati per numero crescente di AS.

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 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. 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. Nel correggere porta **tutte** le strutture relative alla lista in un file list.h.

Per correggere gli errori modifica 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 e provane il funzionamento su una copia di prj1 che devi fare tu.

Metti nel file ~/compito/esercizio2/soluzione2.3.txt i comandi per creare e per applicare la patch. Metti la patch nel file ~/compito/esercizio2/soluzione2.3\_patch.txt.

4. \*\* Crea un Makefile con i seguenti target

- main: crea l'eseguibile linkato staticamente senza simboli di debug

- clean: pulisce il progetto cancellando i file inutili (\*.o, \*.~)

- delete: come clean ma cancella anche i target

- main\_debug: crea eseguibile linkato dinamicamente con simboli di debug

- result.txt: mette l'output dell'esecuzione in result.txt

- test: esegue main e fa un diff tra result.txt e l'output dell'esecuzione mostrando il risultato su standard output (puoi usare un file temporaneo se vuoi).

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

5. \*\* Considera la funzione createList(). Mostra il valore della variabile  $x$  all'inizio della seconda iterazione. Mostra il valore della variabile  $x$  all'inizio della novantesima iterazione.

Considera la lista dopo la creazione. Mostra il contenuto di ciascun campo del penultimo elemento della lista. Mostra anche l'indirizzo di memoria in cui è memorizzata la variabile theList.

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 — A.A. 2005-2006, prova pratica del 28 aprile 2006

## Compito F

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 “~” 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 radb.db.txt contiene un record per ciascun prefisso IP registrato presso un certo registro di Internet. Nel file ciascun record è separato da una linea vuota, i campi sono su linee distinte, tranne il campo "descr" che è su più linee. Per svolgere l'esercizio non è necessario conoscere il significato di tutti i campi. (Suggerimento: il processamento di tale file con awk può essere fatto riga per riga oppure record per record ponendo RS="", cioè stringa vuota, e FS="\n")

1. \* Il campo “changed” riporta alla fine della riga la data dell'ultimo cambiamento. Mostra un comando che dia il numero di prefissi che sono stati cambiati nel primo giorno del mese (=01, mese e anno qualsiasi).

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

2. \*\* Il campo “origin” (sempre presente prima di “changed”) contiene una stringa che identifica l'autonomous system (AS) che origina il prefisso. Considera gli AS che originano almeno un prefisso il cui record è cambiato nel mese di marzo. Mostra un comando che calcoli una tabella con due colonne: una colonna contiene gli AS che originano almeno un prefisso cambiato nel mese di marzo (marzo=03, giorno e anno qualsiasi), l'altra, per ciascun AS, il numero di tali prefissi originati.

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 radb.db.txt e produca su standard output il seguente report:

- i record sono separati da una riga vuota
- esiste un record per ciascun AS originator, tale AS è indicato nella prima riga del record
- le restanti righe del record contengono tutti i prefissi originati da quell'AS
- i record devono essere ordinati per numero crescente di AS.

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 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. 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. Nel correggere porta **tutte** le strutture relative alla lista in un file list.h.

Per correggere gli errori modifica 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 e provane il funzionamento su una copia di prj1 che devi fare tu. Metti nel file ~/compito/esercizio2/soluzione2.3.txt i comandi per creare e per applicare la patch. Metti la patch nel file ~/compito/esercizio2/soluzione2.3\_patch.txt.

4. \*\* Crea un Makefile con i seguenti target

- main: crea l'eseguibile linkato staticamente senza simboli di debug

- clean: pulisce il progetto cancellando i file inutili (\*.o, \*.~)

- delete: come clean ma cancella anche i target

- main\_debug: crea eseguibile linkato dinamicamente con simboli di debug

- result.txt: mette l'output dell'esecuzione in result.txt

- test: esegue main e fa un diff tra result.txt e l'output dell'esecuzione mostrando il risultato su standard output (puoi usare un file temporaneo se vuoi).

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

5. \*\* Considera la funzione createList(). Mostra il valore della variabile  $x$  all'inizio della seconda iterazione. Mostra il valore della variabile  $x$  all'inizio della novantesima iterazione.

Considera la lista dopo la creazione. Mostra il contenuto di ciascun campo del penultimo elemento della lista. Mostra anche l'indirizzo di memoria in cui è memorizzata la variabile theList.

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.