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

Sistemi Operativi — A.A. 2007-2008, prova pratica del 10 aprile 2008

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

Istruzioni per la consegna del compito

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

Esercizio 1

Il file di testo bgp_updates.txt contiene un record per ciascun aggiornamento di routing ricevuto. Nel file ciascun record è separato da una linea vuota, i campi sono su linee distinte, tranne i campi "ANNOUNCE" e "WITHDRAW" che sono su più linee in cui ciascuna linea inizia con due spazi. Per svolgere l'esercizio non è necessario conoscere il significato di tutti i campi. Suggerimenti: alcune volte conviene processare tale file con awk usando RS="" (stringa vuota) e FS="\n", ricorda che, in awk, gsub() e' un efficace strumento di sostituzione.

1. * I campi "ANNOUNCE" e "WITHDRAW" contengono una lista di prefissi. Mostra un comando che selezioni tra tali prefissi quelli che terminano con /24 che hanno il numero 16 come secondo byte (es. 193.**16**.10.0/24).

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

2. ** Dai un comando che mostri una tabella in cui la prima colonna sia il contenuto del campo "NEXT_HOP" e la seconda il contenuto del campo "ASPATH".

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 bgp_updates.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 "NEXT_HOP", tale valore è indicato nella prima riga del record
   - le restanti righe del record contengono tutti gli ASPATH che compaiono in records assieme a quel NEXT_HOP
   - i record devono essere ordinati in modo che NEXT_HOP più frequenti nel file siano in cima.

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 lista contenente una serie di numeri, derivati medinate una funzione, dai 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, “5”.

Per correggere gli errori modifca 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: crea l'eseguibile “fib” linkato dinamicamente con simboli di debug
   - fib_release: crea l'eseguibile “fib” linkato staticamente senza simboli di debug
   - clean: pulisce il progetto cancellando i file inutili (*.o, *.~)
   - delete: come clean ma cancella anche i target
   - dynlib: mostra quali librerie dinamiche vengono linkate all'eseguibile fib

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. ** Considera una esecuzione di fib con parametro 10. Mostra il contenuto dello stack all'inizio della 209-esima esecuzione della funzione fib().
   Considera il momento in cui fib() ritorna il suo valore (il return sull'ultima linea). La prima volta in cui fib() ritorna un valore f >25 (f e' una variabile di fib()), quante volte tale linea e' gia' stata eseguita.

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.

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

Sistemi Operativi — A.A. 2007-2008, prova pratica del 10 aprile 2008

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 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

Istruzioni per la consegna del compito

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

Esercizio 1

Il file di testo bgp_updates.txt contiene un record per ciascun aggiornamento di routing ricevuto. Nel file ciascun record è separato da una linea vuota, i campi sono su linee distinte, tranne i campi "ANNOUNCE" e "WITHDRAW" che sono su più linee in cui ciascuna linea inizia con due spazi. Per svolgere l'esercizio non è necessario conoscere il significato di tutti i campi. Suggerimenti: alcune volte conviene processare tale file con awk usando RS="" (stringa vuota) e FS="\n", ricorda che, in awk, gsub() e' un efficace strumento di sostituzione.

1. * I campi "ANNOUNCE" e "WITHDRAW" contengono una lista di prefissi. Mostra un comando che selezioni tra tali prefissi quelli che terminano con "/24" che hanno il numero 23 come terzo byte (es. 193.43.**23**.0/24).

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

2. ** Dai un comando che mostri una tabella in cui la prima colonna sia il contenuto del campo "TO" e la seconda il contenuto del campo "COMMUNITY".

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 bgp_updates.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 "NEXT_HOP", tale valore è indicato nella prima riga del record
   - le restanti righe del record contengono tutti gli ASPATH che compaiono in records assieme a quel NEXT_HOP
   - i record devono essere ordinati in modo che NEXT_HOP più frequenti nel file siano in cima.

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 lista contenente una serie di numeri, derivati medinate una funzione, dai 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, “5”.

Per correggere gli errori modifca 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: crea l'eseguibile “fib” linkato dinamicamente con simboli di debug
   - fib_release: crea l'eseguibile “fib” linkato staticamente senza simboli di debug
   - clean: pulisce il progetto cancellando i file inutili (*.o, *.~)
   - delete: come clean ma cancella anche i target
   - dynlib: mostra quali librerie dinamiche vengono linkate all'eseguibile fib

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. ** Considera una esecuzione di fib con parametro 10. Mostra il contenuto dello stack all'inizio della 209-esima esecuzione della funzione fib().
   Considera il momento in cui fib() ritorna il suo valore (il return sull'ultima linea). La prima volta in cui fib() ritorna un valore f >25 (f e' una variabile di fib()), quante volte tale linea e' gia' stata eseguita.

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.

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

Sistemi Operativi — A.A. 2007-2008, prova pratica del 10 aprile 2008

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 "~" 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

Istruzioni per la consegna del compito

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

Esercizio 1

Il file di testo bgp_updates.txt contiene un record per ciascun aggiornamento di routing ricevuto. Nel file ciascun record è separato da una linea vuota, i campi sono su linee distinte, tranne i campi "ANNOUNCE" e "WITHDRAW" che sono su più linee in cui ciascuna linea inizia con due spazi. Per svolgere l'esercizio non è necessario conoscere il significato di tutti i campi. Suggerimenti: alcune volte conviene processare tale file con awk usando RS="" (stringa vuota) e FS="\n", ricorda che, in awk, gsub() e' un efficace strumento di sostituzione.

1. * I campi "ANNOUNCE" e "WITHDRAW" contengono una lista di prefissi. Mostra un comando che selezioni tra tali prefissi quelli che terminano con /24 che hanno il numero 13 come secondo byte (es. 193.**13**.1.0/24).

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

2. ** Dai un comando che mostri una tabella in cui la prima colonna sia il contenuto del campo "NEXT_HOP" e la seconda il contenuto del campo "TO".

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 bgp_updates.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 "NEXT_HOP", tale valore è indicato nella prima riga del record
   - le restanti righe del record contengono tutti gli ASPATH che compaiono in records assieme a quel NEXT_HOP
   - i record devono essere ordinati in modo che NEXT_HOP più frequenti nel file siano in cima.

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 lista contenente una serie di numeri, derivati medinate una funzione, dai 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, “5”.

Per correggere gli errori modifca 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: crea l'eseguibile “fib” linkato dinamicamente con simboli di debug
   - fib_release: crea l'eseguibile “fib” linkato staticamente senza simboli di debug
   - clean: pulisce il progetto cancellando i file inutili (*.o, *.~)
   - delete: come clean ma cancella anche i target
   - dynlib: mostra quali librerie dinamiche vengono linkate all'eseguibile fib

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. ** Considera una esecuzione di fib con parametro 10. Mostra il contenuto dello stack all'inizio della 209-esima esecuzione della funzione fib().
   Considera il momento in cui fib() ritorna il suo valore (il return sull'ultima linea). La prima volta in cui fib() ritorna un valore f >25 (f e' una variabile di fib()), quante volte tale linea e' gia' stata eseguita.

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.

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

Sistemi Operativi — A.A. 2007-2008, prova pratica del 10 aprile 2008

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 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

Istruzioni per la consegna del compito

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

Esercizio 1

Il file di testo bgp_updates.txt contiene un record per ciascun aggiornamento di routing ricevuto. Nel file ciascun record è separato da una linea vuota, i campi sono su linee distinte, tranne i campi "ANNOUNCE" e "WITHDRAW" che sono su più linee in cui ciascuna linea inizia con due spazi. Per svolgere l'esercizio non è necessario conoscere il significato di tutti i campi. Suggerimenti: alcune volte conviene processare tale file con awk usando RS="" (stringa vuota) e FS="\n", ricorda che, in awk, gsub() e' un efficace strumento di sostituzione.

1. * I campi "ANNOUNCE" e "WITHDRAW" contengono una lista di prefissi. Mostra un comando che selezioni tra tali prefissi quelli che terminano con /24 che hanno il numero 1 come terzo byte (es. 193.34.**1**.0/24).

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

2. ** Dai un comando che mostri una tabella in cui la prima colonna sia il contenuto del campo "COMMUNITY" e la seconda il contenuto del campo "ASPATH".

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 bgp_updates.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 "NEXT_HOP", tale valore è indicato nella prima riga del record
   - le restanti righe del record contengono tutti gli ASPATH che compaiono in records assieme a quel NEXT_HOP
   - i record devono essere ordinati in modo che NEXT_HOP più frequenti nel file siano in cima.

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 lista contenente una serie di numeri, derivati medinate una funzione, dai 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, “5”.

Per correggere gli errori modifca 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: crea l'eseguibile “fib” linkato dinamicamente con simboli di debug
   - fib_release: crea l'eseguibile “fib” linkato staticamente senza simboli di debug
   - clean: pulisce il progetto cancellando i file inutili (*.o, *.~)
   - delete: come clean ma cancella anche i target
   - dynlib: mostra quali librerie dinamiche vengono linkate all'eseguibile fib

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. ** Considera una esecuzione di fib con parametro 10. Mostra il contenuto dello stack all'inizio della 209-esima esecuzione della funzione fib().
   Considera il momento in cui fib() ritorna il suo valore (il return sull'ultima linea). La prima volta in cui fib() ritorna un valore f >25 (f e' una variabile di fib()), quante volte tale linea e' gia' stata eseguita.

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.