

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

Sistemi Operativi — A.A. 2007-2008, prova pratica del 25 settembre 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 “`~`” significa “home dell'utente”).
- Gli esercizi sono etichettati con 1,2 o 3 asterischi:
 - * = esercizio semplice, risolvi questi prima degli altri
 - ** = esercizio di media difficoltà
 - *** = esercizio difficile, 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 `ibgplay.access.log` contiene un record per ciascun accesso ad un web server, i campi sono delimitati da spazi tranne quando esplicitamente indicato. Il campo 1 contiene l'indirizzo IP della macchina che ha fatto l'accesso. Ignora i campi 2 e 3 (quasi sempre pari a “-”). Il campo 4 (delimitato da parentesi quadre) mostra l'istante di accesso. Il campo 5 mostra la richiesta fatta ed è una stringa delimitata da doppi apici che può contenere degli spazi. La prima parola di tale campo è GET, POST, o altro metodo HTTP. Il campo 6 è un codice di errore. Ignora i campi 7 e 8. Il campo 9 è una stringa delimitata da doppi apici che può contenere degli spazi e fornisce informazioni sul browser.

1. * Mostra un comando che selezioni le righe che hanno nel campo 1 un ip con il terzo numero pari a 161 e siano relative a richieste **POST** effettuate tra le 10:00 e le 10:59.
Scrivi nel file `~/compito/esercizio1/soluzione1.1.txt` il comando usato e il suo output (fai copia-e-incolla dal terminale).
2. ** Il campo 5 (delimitato da doppi apici) contiene la *richiesta* fatta dal client. Crea un comando che dia una tabella **quantità ip richiesta** che mostri quante volte un certo *ip* ha fatto una certa richiesta. La tabella deve essere in ordine **decrescente** di quantità e contenere i **top 15** elementi.
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 `ibgplay.access.log` e produca su standard output il seguente report:
 - i record sono separati da una riga vuota
 - esiste un record per ciascun valore della *richiesta* (campo 5), tale valore è indicato nella prima riga del record
 - le restanti righe del record contengono tutti i valori della coppia *ip, istante di accesso* (campo 4 tra quadre), eventualmente duplicati, che compaiono in record assieme a quella *richiesta*
 - i record devono essere ordinati in modo che *richieste* 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 inizializza una lista contenente una serie di numeri, derivati mediante una funzione, dai primi n numeri di fibonacci e ne stampa il contenuto (n è passato come parametro). La directory prj2 contiene una copia di prj1 che devi modificare per rispondere alle seguenti domande.

1. * Usa il comando "gcc -Wall *.c" per una prima compilazione (l'opzione -Wall attiva tutti i warning). Correggi gli errori/warning di pre-processamento, 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 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: crea l'eseguibile "fib" linkato dinamicamente con simboli di debug e compilazione con tutti i warning
 - fib_release: crea l'eseguibile "fib_release" linkato staticamente senza simboli di debug e compilazione ottimizzata (vedi man gcc)
 - clean: pulisce il progetto cancellando i file inutili (*.o, *.~)
 - delete: come clean ma cancella anche i target
 - main.i: risultato della precompilazione del file main.c

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

3. * Crea una patch tra prj1 e prj2 e provala su una copia "prj1 copy" 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 11. Mostra il contenuto dello stack all'inizio della 301-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 $f > 50$ (f è una variabile di fib()), quante volte tale linea è già 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 25 settembre 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 “~” significa “home dell'utente”).
- Gli esercizi sono etichettati con 1,2 o 3 asterischi:
 - * = esercizio semplice, risolvi questi prima degli altri
 - ** = esercizio di media difficoltà
 - *** = esercizio difficile, 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 `ibgplay.access.log` contiene un record per ciascun accesso ad un web server, i campi sono delimitati da spazi tranne quando esplicitamente indicato. Il campo 1 contiene l'indirizzo IP della macchina che ha fatto l'accesso. Ignora i campi 2 e 3 (quasi sempre pari a "-"). Il campo 4 (delimitato da parentesi quadre) mostra l'istante di accesso. Il campo 5 mostra la richiesta fatta ed è una stringa delimitata da doppi apici che può contenere degli spazi. La prima parola di tale campo è GET, POST, o altro metodo HTTP. Il campo 6 è un codice di errore. Ignora i campi 7 e 8. Il campo 9 è una stringa delimitata da doppi apici che può contenere degli spazi e fornisce informazioni sul browser.

1. * Mostra un comando che selezioni le righe che hanno nel campo 1 un ip con il terzo numero pari a **80** e siano relative a richieste **GET** effettuate tra le 12:00 e le 12:59.

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

2. ** Il campo 9 (delimitato da doppi apici) contiene il *browser* usato dal client. Crea un comando che dia una tabella **quantità ip browser** che mostri quante volte un certo *ip* ha fatto una richiesta con un certo *browser*. La tabella deve essere in ordine **decrescente** di quantità e contenere i **top 15** elementi.

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 `ibgplay.access.log` e produca su standard output il seguente report:

- i record sono separati da una riga vuota
- esiste un record per ciascun valore del *browser* (campo 9), tale valore è indicato nella prima riga del record
- le restanti righe del record contengono tutti i valori della coppia *ip, istante di accesso* (campo 4 tra quadre), eventualmente duplicati, che compaiono in record assieme a quel *browser*
- i record devono essere ordinati in modo che *browser* 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 inizializza una lista contenente una serie di numeri, derivati mediante una funzione, dai primi n numeri di fibonacci e ne stampa il contenuto (n è passato come parametro). La directory prj2 contiene una copia di prj1 che devi modificare per rispondere alle seguenti domande.

1. * Usa il comando "gcc -Wall *.c" per una prima compilazione (l'opzione -Wall attiva tutti i warning). Correggi gli errori/warning di pre-processamento, 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 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: crea l'eseguibile "fib" linkato dinamicamente con simboli di debug e compilazione con tutti i warning
 - fib_release: crea l'eseguibile "fib_release" linkato staticamente senza simboli di debug e compilazione ottimizzata (vedi man gcc)
 - clean: pulisce il progetto cancellando i file inutili (*.o, *.~)
 - delete: come clean ma cancella anche i target
 - main.i: risultato della precompilazione del file main.c

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

3. * Crea una patch tra prj1 e prj2 e provala su una copia "prj1 copy" 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 11. Mostra il contenuto dello stack all'inizio della 301-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 $f > 50$ (f è una variabile di fib()), quante volte tale linea è già 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.