

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

Sistemi Operativi — A.A. 2005-2006, prova pratica del 7 luglio 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 access.log contiene un record per ciascun accesso ad un web server. Il campo 1 contiene l'indirizzo IP della macchina che ha fatto l'accesso. I campi 2 e 3 sono sempre pari a “-”, ignorali. Il campo 4 (delimitato da parentesi quadre) mostra l'istante di accesso. Il campo 5 è una stringa delimitata da doppi apici e 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 altre informazioni.

1. * Conta tutte le righe in cui il codice di errore (campo 6) è pari a 200 e il metodo di accesso è POST. Scrivi nel file ~/compito/esercizio1/soluzione1.1.txt il comando usato e il suo output (fai copia-e-incolla dal terminale).
2. ** Considera il campo 5. Mostra un comando che calcoli una tabella con due colonne: una colonna contiene tutti i possibili valori che il campo 5 assume nel file e l'altra colonna il numero di volte che tale valore appare nel campo 5. Ordina la tabella secondo quest'ultima colonna. (Suggerimento: per gestire i campi delimitati da doppi apici puoi usare la variabile FS di awk ponendola pari a “”).
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 access.log e produca su standard output il seguente report:
 - i record sono separati da una riga vuota
 - esiste un record per ciascun indirizzo IP che origina una richiesta, tale IP è indicato nella prima riga del record
 - le restanti righe del record contengono le righe di log per gli accessi fatti da tale IP omettendo il primo campo che è l'IP stesso.
 - i record devono essere ordinati per quantità di richieste (prima quelli degli IP che hanno fatto meno richieste)

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 list.h. Mostra quindi l'output dell'esecuzione del programma corretto.

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

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
 - result.txt: mette l'output dell'esecuzione di main in result.txt.
 - result_debug.txt: mette l'output dell'esecuzione di main_debug in result_debug.txt.
 - diff: mostra il diff tra result.txt e result_debug.txt su standard output.
 - clean: pulisce il progetto cancellando i file inutili (*.o, *.~)
 - delete: come clean ma cancella anche i target

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

4. ** Considera la funzione crea(). Mostra il valore della variabile x all'inizio della terza iterazione. Mostra lo stack delle chiamate. Mostra come il valore della variabile x cambia durante l'esecuzione per $i=40$.

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.

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

Sistemi Operativi — A.A. 2005-2006, prova pratica del 7 luglio 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 xvfz 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 access.log contiene un record per ciascun accesso ad un web server. Il campo 1 contiene l'indirizzo IP della macchina che ha fatto l'accesso. I campi 2 e 3 sono sempre pari a “-”, ignorali. Il campo 4 (delimitato da parentesi quadre) mostra l'istante di accesso. Il campo 5 è una stringa delimitata da doppi apici e 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 altre informazioni.

1. * Conta tutte le righe in cui il codice di errore (campo 6) è pari a 404 e il metodo di accesso è GET. Scrivi nel file ~/compito/esercizio1/soluzione1.1.txt il comando usato e il suo output (fai copia-e-incolla dal terminale).
2. ** Considera il campo 9. Mostra un comando che calcoli una tabella con due colonne: una colonna contiene tutti i possibili valori che il campo 9 assume nel file e l'altra colonna il numero di volte che tale valore appare nel campo 9. Ordina la tabella secondo quest'ultima colonna. (Suggerimento: per gestire i campi delimitati da doppi apici puoi usare la variabile FS di awk ponendola pari a “”). 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 access.log e produca su standard output il seguente report:
 - i record sono separati da una riga vuota
 - esiste un record per ciascun indirizzo IP che origina una richiesta, tale IP è indicato nella prima riga del record
 - le restanti righe del record contengono le righe di log per gli accessi fatti da tale IP omettendo il primo campo che è l'IP stesso.
 - i record devono essere ordinati per quantità di richieste (prima quelli degli IP che hanno fatto meno richieste)
4. 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 list.h. Mostra quindi l'output dell'esecuzione del programma corretto.

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

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
 - result.txt: mette l'output dell'esecuzione di main in result.txt.
 - result_debug.txt: mette l'output dell'esecuzione di main_debug in result_debug.txt.
 - diff: mostra il diff tra result.txt e result_debug.txt su standard output.
 - clean: pulisce il progetto cancellando i file inutili (*.o, *.~)
 - delete: come clean ma cancella anche i target

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

4. ** Considera la funzione crea(). Mostra il valore della variabile x all'inizio della terza iterazione. Mostra lo stack delle chiamate. Mostra come il valore della variabile x cambia durante l'esecuzione per $i=40$.

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.