

GRUPPI

Qualche studente NON risulta ancora inserito in un gruppo: TUTTI QUELLI CHE SONO IN QUESTA SITUAZIONE potrebbero indicarlo chiaramente nel foglio di iscrizioni online, lasciando un recapito (per esempio una mail temporanea o quella d'ateneo) per poter essere contattati dagli altri.

Il giorno 10/05 la lista online sarà considerata "chiusa".

SUPPORTO

Tutti i giovedì di Maggio (8, 15, 22, 29) 2014 dalle 10.00 alle 12.00 in Aula B106 (PER ENTRAMBI I GRUPPI)

PROGETTI: segnalazioni/correzioni/precisazioni (ovviamente - come già spiegato - soluzioni più complesse, se funzionanti, possono essere valutate positivamente)

- nelle regole del "make" usare "RES" al posto di "ASSETS": la "generazione" dei file di input può utilizzare contenuti "pre-confezionati" che di fatto saranno solo copiati in output; come plus-valore si può realizzare un algoritmo per generare dinamicamente dei contenuti di input
- il "make test" deve effettuare delle chiamate di prova complete (anche più di una, è molto gradito, su input differenti che possono essere anche generati "al volo": formattare opportunamente l'output soprattutto in questo caso evidenziando ad esempio "ESECUZIONE TEST #01", "ESECUZIONE TEST #02" e così via...): si può renderlo parzialmente interattivo se i test sono diversi (ad esempio mostrando una breve descrizione di che operazione si sta per eseguire ed attendendo la pressione di un tasto per proseguire) se desiderato
- per la gestione dei parametri da riga di comando è valutato positivamente l'uso di getopt e di opzioni in formato standard (singolo/doppio dash)

Progetto #01

- nella codifica di esempio ci sono un paio di errori di trascrizione per cui il messaggio codificato è leggermente sbagliato (!)
- i comandi di "interazione" del server possono avvenire a video (ad esempio con un menu: in questo caso si potrebbero aprire più "console" sulla VM attraverso l'uso di connessioni SSH), ma anche tramite chiamate come già per le funzioni di base (e per questo scopo si possono aggiungere nuovi parametri per server e/o client o anche realizzare un terzo programma tipo "codecservctrl")
- per le opzioni della riga di comando si possono seguire le regole di default con singoli dash per opzioni corte e doppi dash per quelle lunghe (es. "-n ..." / "-name ...")

- nell'esempio di chiamata del client l'opzione ipotetica "-file" serve a specificare che quella "-message" riferisce un file, se non ci fosse (o magari ci fosse un'alternativa tipo "-text") allora "-message" riferirebbe un testo direttamente, si possono usare anche altre soluzioni
- la codifica mantiene la capitalizzazione (se la prima lettera del messaggio originale è maiuscola, lo è anche quella della versione codificata e viceversa): tutti i caratteri al di fuori del set gestito possono essere lasciati non codificati (come per gli spazi); durante la codifica/decodifica conta solo la posizione della lettera all'interno dell'alfabeto
- il "make test" dovrà almeno lanciare un server e un client

Progetto #02

- i processi che rappresentano partite per cui non si sono "dipendenze" possono essere eseguiti contemporaneamente (al limite anche far giocare alla stessa squadra più partite contemporaneamente)
- usare come unità di tempo millisecondi anziché secondi, impostando un valore di default ragionevole (e magari anche dei limiti minimo/massimo) se l'utente non inserisce alcun valore
- i lassi di tempo inferiori ai range considerati si possono "normalizzare": ad esempio in un tempo di 45' con 7' di recupero si possono considerare quattro frazioni di 15' + 15' + 15' + 7' (quest'ultima da considerarsi come di 15' ai fini dei calcoli probabilistici)
- TYPO: "classifiche dei giorni" -> "classifiche dei gironi" (le classifiche servono per i vari gironi, NON per singola giornata)
- il "make test" dovrà almeno lanciare un'esecuzione

Progetto #03

- l'ordinamento per "id" può usare quello che compare con "Ismod" ("Used by")
- il "make test" dovrà almeno lanciare un'esecuzione di ogni comando realizzato