

Progetto di Sistemi Operativi Modulo II (Canale 1) A.A 2011/2012.

Obiettivo di questo progetto è di definire tre programmi per la gestione di un sistema di esecuzione di comandi in ambiente controllato.

In particolare deve essere definito un programma (*requester*) che sottomette (scrivendoli in una fifo) i comandi da eseguire ad un esecutore (*executor*) ed un terzo programma (*monitor*) che supervisiona l'attività dell'*executor*.

Le funzioni dei programmi sono le seguenti:

- **Monitor:** lancia il programma *executor*, scrive su file di log gli eventi principali. Attende che l'*executor* termini l'esecuzione e:
 1. Scrive su file di log lo status di terminazione dell'*executor*.
 2. Scrive su file di log il numero di comandi eseguiti con successo e quanti con errore da parte dell'*executor*.
 3. Termina l'esecuzione.
- **Executor:**
 - a. Acquisisce come parametro (*jail_dir*) un path di directory e il path di un file di tipo fifo (*fifo_path*).
 - b. Modifica la propria root directory in *jail_dir*.
 - c. [FACOLTATIVO: Presupponendo che il programma sia *setuid* a root ed eseguito da un utente diverso da root] Abbassare i privilegi impostandoli ai privilegi dell'utente che ha invocato il programma.
 - d. Legge da una fifo, uno alla volta, i comandi inviatigli dal requester e li esegue (se diversi da quelli specificati al punto successivo) notificando (tramite invio di segnali) al monitor l'errore o la corretta esecuzione.
 - e. Termina la propria esecuzione quando riceve il comando 'shutdown' (termina con successo) o 'abort' (termina con errore).
 - f. L'*executor* deve poter essere terminato anche tramite invio di opportuni segnali (sceglierne almeno due diversi).
- **Requester:** Programma client utilizzato dall'utente per sottomettere i comandi all'*executor*.

Definire inoltre uno script bash (*monitor-wrapper.sh*) per l'esecuzione del programma *monitor*. Lo script serve da wrapper per invocare il programma *monitor* e deve individuare (nell'intero filesystem ad esclusione di */dev/*, */cdrom* e */proc* e per una profondità di massimo di 3 directory) la directory *LabSOMod2Canale1* ed eseguire il programma *monitor* ivi contenuto.

La compilazione, installazione e disinstallazione del progetto deve essere gestita con il comando *make*. È quindi indispensabile definire un opportuno file *Makefile*.

Dettagli implementativi e vincoli

1. Il progetto può essere sviluppato in gruppo (max 3 persone). Ad ogni gruppo viene assegnato un identificativo dato dal cognome del primo (in senso alfabetico) membro del gruppo seguito da '2012' (es: alamo2012, rossi2012,...). In caso di omonimia si aggiunge il nome (es: alamoclaudio2012, alamoemanuela2012).
2. Ogni membro del gruppo deve poter essere in grado di discutere ed argomentare **QUALSIASI** aspetto del progetto.
3. Il progetto deve essere sviluppato su sistema Linux e prodotto utilizzando le tecnologie e nozioni (es: linguaggio C e programmazione di sistema) affrontate al corso.
4. Il progetto DEVE essere compilabile ed eseguibile sui sistemi Linux installati al laboratorio Colossus del Dipartimento di Informatica (Via Salaria 113).
5. È INDISPENSABILE (pena la non correzione del progetto) rispettare ALMENO le seguenti regole di sviluppo:
 - a. Indentare il codice. (e verificare che l'indentazione sia mantenuta coerente visualizzando il file tramite editor tipo vi o nano).
 - b. Non utilizzare costanti nel codice ma definire delle MACRO
 - c. Definire per ogni file sorgente .c un relativo file di definizione .h in cui definire prototipi di funzione, macro e tipi di dati.

Package da consegnare.

La consegna del progetto deve avvenire almeno 7 giorni prima della data dell'appello a cui si vuole partecipare.

L'archivio deve contenere tutto il necessario per la compilazione, esecuzione e test del progetto, incluso la documentazione che deve essere in formato pdf o OpenOffice. Inoltre deve contenere anche (ALMENO) due script (denominati `test1.sh` e `test2.sh`) che automatizzano il test dell'applicativo (ogni script deve eseguire in maniera automatizzata un singolo caso di test).

La relazione deve contenere (almeno) due capitoli:

- **Scelte progettuali:** In questo capitolo vengono illustrati i principali problemi affrontati e le scelte organizzative/architetturali ed implementative effettuate.
- **Contributi ed Organizzazione:** in cui si descrive, in paragrafi distinti:
 - La lista dei membri del gruppo (nome cognome e matricola).
 - Istruzioni per la compilazione ed esecuzione del/dei programmi.
 - Gli esempi di test (implementati con `test1.sh` e `test2.sh`).

La consegna deve consistere in un package, da spedire via mail:

- **subject:** **SOMIIC1 Pro_2012: cognome1, cognome2, ... cognome3**)
- **allegato:** Progetto in formato tar compresso col nome **executor_0.1.tgz**. (non deve contenere file producibili con l'uso del comando make e che siano prodotto di esecuzione (es: log files)).
- **Body:** nel corpo della mail dovete inserire (su righe separate) il nome, cognome e matricola di ogni membro del gruppo.
- **To:** gabrielli@di.uniroma1.it