

**Funzionalità:**

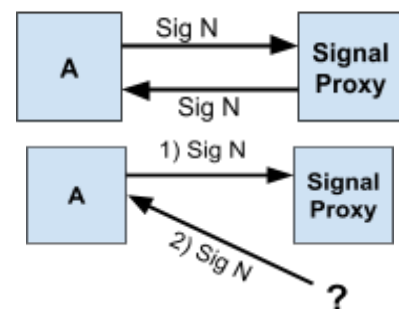
Viene richiesta la creazione di un programma (*SignalProxy*) in grado di soddisfare determinate richieste e requisiti. L'implementazione delle funzionalità, e le tecnologie da esse impiegate, sono a discrezione dell'esaminato. Nel seguente testo di esame, per *SignalProxy* ci si riferisce al processo generato all'esecuzione del binario *SignalProxy.out*. Nel resto del testo, ci si riferirà ad un Processo A come ad un qualunque altro processo eseguito sul sistema.

**VINCOLI e ACCORGIMENTI:**

- La cartella finale deve contenere uno o più file sorgente (almeno il "main.c") ed eventualmente altri header .h ed un Makefile "Makefile". Nessun altro tipo di file, sia esso un eseguibile o un file testuale, è ammesso.
- Qualora non sia presente un Makefile, l'applicazione deve essere compilabile tramite un unico file main.c.
- La sintassi dei comandi deve essere rispettata. In caso di discrepanze il punteggio attribuito sarà nullo.
- Sono concessi output di debugging a discrezione dell'esaminato.
- Il codice consegnato deve essere commentato spiegando le varie funzionalità ed i vari passaggi.

**ATTIVITÀ:**

- **[3/30] Makefile:** realizzare correttamente il "Makefile" e far sì che digitando "make FILE=<path>" venga creato il binario *SignalProxy* (partendo dai file sorgente nella cartella). Inoltre, se il file <path> non esiste, esso deve essere creato dal Makefile contenente la scritta "start↵". Qualora il file esistesse già, il makefile deve semplicemente creare il binario, senza modificare il contenuto del file. Creare infine una regola "clean" per la rimozione del binario generato e del file <path>.
- Creare un programma in C *SignalProxy* che accetti esattamente un parametro <pathToLogFile> (quindi che possa essere lanciato con ". /SignalProxy <pathToLogFile>"). E che gestisca le seguenti funzionalità:
  - **[5/30] Repeater:** se un processo A invia un segnale SIGUSR1 al processo *SignalProxy*, il processo A deve ricevere subito un segnale SIGUSR1 dallo stesso processo contattato da A.
  - **[5/30] Relay di un segnale:** se un processo A invia un segnale SIGUSR2 al processo *SignalProxy*, il processo A deve subito ricevere lo stesso segnale SIGUSR2 da un processo B qualunque, diverso da quello contattato da A.
  - **[6/30] Signal Logger:** se un processo A invia un segnale SIGUSR1 o SIGUSR2 a *SignalProxy*, il file *logFile* viene aggiornato **dopo 3 secondi** (append) con la dicitura "#pidA-#signNo↵" dove, #pidA è il PID del processo A, #signNo è il numero del segnale inviato. Al comando <CTRL-C> il programma deve scrivere "stop↵" in fondo al file (scritto subito senza attendere). Durante i 3 secondi di ritardo, il programma deve essere in grado di rispondere a qualunque altra richiesta. Si può assumere che il file esista già (per esempio è stato creato manualmente o tramite makefile).
  - **[6/30] Queue checker:** Se un processo A scrive un messaggio sulla coda associata alla chiave generata dalla coppia (<logFile>, 1) il PID di un processo "#noPid", *SignalProxy*



deve occuparsi, in qualche modo, di far ricevere al processo con il PID = `#noPid` un segnale `SIGALRM`. Si può assumere che la coda esista e che il PID comunicato abbia una sintassi corretta e corrisponda ad un processo esistente.

- **[5/30] Bash script:** Creare uno script bash che riceva come parametro il nome di un file avente la stessa sintassi del logfile sopra citata, e restituisca a video tutte le righe, stampando in maniera alternata su *stderr* e *stdout* (con la prima riga stampata su *stdout*).