LabSO

Progetti

Progetto 1: shell custom con piping

- **Scrivere una "shell custom" che all'avvio accetta alcuni parametri, tra cui i nomi di due file di log: uno per "out" ed un per "err". La shell deve accettare comandi con parametri e almeno il carattere speciali di piping e il codice di ritorno. Tutto l'output di "out" ed "err" deve andare, oltre che nei canali standard, ANCHE sui file di log impostati all'avvio e ben formattato. Ad esempio eseguendo da dentro la shell ls -l | wc si dovrebbe avere a video il conteggio delle statistiche, ma gli stessi dati dovrebbero anche essere salvati su file in forma completa e/o sintetica, anche con i codici di ritorno.**

Progetto 2: statistiche comandi e analisi

- **Implementare un tool che esegue comandi esterni del tipo run "<cmd> <args>" (ad es.: run "ls -l | wc" richiama il comando "ls -l | wc"...) gestendo le statistiche d'uso: per ogni esecuzione tiene traccia del tempo d'esecuzione salvando su un file di "log" le informazioni principali compreso il codice di ritorno. Si richiede che il “logger” sia un processo autonomo “unico” (quindi all’esecuzione si crea con un “fork” se non esiste o si “utilizza” quello già esistente: ad esso vanno inviate le informazioni da gestire). Il “logger” potrebbe essere un “demone”, ma la concorrenza si avrebbe comunque lanciando il tool da più shell differenti.**

Progetto 3: I/O a 8 bit su Raspberry

- **Il “cartone delle uova” IoT su Raspberry**
- **Un cartone delle uova da 6 posti ha un sensore in ogni spazio che passa dal segnale "0" quando è vuoto a "1" quando vi si poggia un uovo e viceversa quando lo si toglie. Tali sensori sono numerati da 0 a 5. Esistono poi altri due indicatori di input (sensori 6 e 7) che indicano quante uova (da 0 a 3) sono come minimo presenti in magazzino.**
- **In output sono presenti i sensori da 0 a 2 che devono rappresentare il numero di uova presenti nel cartone (la configurazione 1.1.1 non è utilizzata), i sensori 3 e 4, normalmente a 0, che se settati comandano di spostare la quantità di uova indicata dal magazzino al cartone - si resettano poi "automaticamente" - e quelli da 5 a 7 che indicano quante uova si vogliono ordinare (anche qui la configurazione 1.1.1 non è utilizzata).**
- **Il programma deve monitorare i bit di input e in base ai dati presenti aggiornare quelli di output: impostando i bit da 0 a 2 con la quantità attuale e "ordinando" altre uova in modo da cercare di tenere il cartone sempre pieno. Quando in magazzino sono presenti uova devono essere spostate nel cartone.**
- **Si deve fare un processo per gestire ogni "bit", uno per i gruppi e uno generale di gestione.**
- **In mancanza di hardware serve anche un "simulatore" che - tramite ad esempio input da tastiera - simuli almeno il prelievo di uova dal cartone e i vari stati di I/O**

  nota: la quantità in magazzino è in effetti da 0 a 5, ma I sensori indicano valori da 0 a 3. Se il cartone si svuotasse completamente si potrebbero ordinare 5 uova, quindi trasferirne 3 e trovare comunque subito una disponibilità di 2 nuovamente spostabili