

Laboratorio di Sistemi Operativi

Progetto A.A. 2019/2020- Fly-by-Wire: LabSO edition

1. Scopo e fondamenti del progetto

Obiettivo del progetto è emulare un'architettura, anche se enormemente minimale e difforme, di un sistema Fly-By-Wire. Al fine di contestualizzare il progetto e alcuni termini, si precisa che si è utilizzato come riferimento l'architettura del Boeing 777 secondo la descrizione in:

Yeh, Ying C. "Design considerations in Boeing 777 fly-by-wire computers." Proceedings Third IEEE International High-Assurance Systems Engineering Symposium (Cat. No. 98EX231). IEEE, 1998.

I requisiti del progetto sono descritti nel seguito del testo. Dove non specificato con il termine *facoltativo*, i requisiti definiti hanno carattere prescrittivo, con significato equivalente alla parola MUST secondo l'RFC 2119.

2. Descrizione del sistema da sviluppare

Il sistema comprende vari componenti, che mostrano una specifica funzionale ed interazioni secondo quanto descritto di seguito. I componenti si dividono in PFC, Transducers, Generatore Fallimenti, WES, PFC Disconnect Switch.

Ciascun componente è rappresentato da almeno un processo, secondo quanto descritto di seguito.

PFC. Questo componente è realizzato da tre processi PFC1, PFC2, PFC3, che svolgono le medesime funzioni. Una volta al secondo, ciascun PFC1, PFC2, PFC3:

- Acquisisce una nuova stringa NMEA GPGLL dal file G18.txt, e da questa estrae latitudine e longitudine;
- Calcola la distanza percorsa, rispetto alle coordinate acquisite all'istante precedente, e conseguentemente calcola la velocità attuale (all'avvio, ovvero per il primo accesso al file G18.txt, la velocità è 0);
- Comunica la velocità elaborata al processo Transducers.

Il processo PFC1 comunica con Transducers tramite una socket, il processo PFC2 comunica tramite una pipe, ed il processo PFC3 comunica tramite scrittura su un file condiviso.

Inoltre, PFC1, PFC2, PFC3 devono essere in grado di ricevere ed elaborare i segnali inviati dal Generatore Fallimenti, descritto di seguito.

Generatore Fallimenti. Una volta al secondo, questo processo seleziona in modo casuale un processo tra PFC1, PFC2 o PFC3. Quindi:

- Con probabilità 10^{-2} , invia il segnale SIGSTOP al processo in questione
- Con probabilità 10^{-4} , invia il segnale SIGINT al processo in questione
- Con probabilità 10^{-1} , invia il segnale SIGCONT al processo in questione
- Con probabilità 10^{-1} , invia il segnale SIGUSR1 che altera il valore del prossimo calcolo della velocità, effettuando un left shift di 2 bits della velocità calcolata una volta arrotondata all'intero più vicino.
- Effettua infine il log dell'azione su un file *failures.log*

Nota: si possono verificare anche più di uno degli eventi di sopra, per lo stesso istante di tempo.

Transducers. Una volta al secondo, questo processo (o più processi) acquisisce la velocità inviata da PFC1, PFC2, PFC3, ed effettua il log dei valori ricevuti rispettivamente nei file *speedPFC1.log*, *speedPFC2.log*, *speedPFC3.log*.

WES. Una volta al secondo, questo processo accede ai valori di velocità registrati su *speedPFC1.log*, *speedPFC2.log*, *speedPFC3.log*. Se i tre valori sono concordi, invia sullo standard output un messaggio di OK. Se due valori su tre sono concordi, ed un valore è discorde, invia un messaggio di ERRORE al PFC Disconnect Switch, indicando il processo che ha inviato il valore discorde. Se invece i tre valori sono discordi, invia un messaggio di EMERGENZA al PFC Disconnect Switch.

Tutti i messaggi del WES sono visualizzati sullo standard output e inseriti in un file di log *status.log*

PFC Disconnect Switch. Questo processo, quando riceve dal WES un messaggio di ERRORE, controlla lo stato del processo coinvolto.

Quando invece PFC Disconnect Switch riceve dal WES un messaggio di EMERGENZA, PFC Disconnect Switch termina l'intera applicazione.

Tutte le azioni di PFC Disconnect Switch sono registrate nel file di log *switch.log*.

Facoltativo: PFC Disconnect Switch sblocca il processo se bloccato, e lo riavvia altrimenti. In entrambi i casi, il processo in questione deve riprendere a leggere dal punto giusto del file *G18.txt*.

3. Note implementativi del programma richiesto.

Ciascun componente deve essere rappresentato da almeno **un processo**. Non sono ammessi thread. E' ammessa l'esecuzione come root, se necessaria (da motivare nella relazione).

4. Modalità di avvio ed esecuzione.

Il programma è avviato con un singolo comando di shell, indicando il percorso del file *G18.txt* come parametro.

Esempio. L'avvio del programma potrebbe essere la riga: `$eseguibile /home/ceccarelli/labso/G18.txt`

L'esecuzione termina quando l'ultima stringa GPGLL del file *G18.txt* è elaborata dalla PFC, o a seguito del messaggio EMERGENZA. Al termine dell'esecuzione, tutti i processi del programma dovranno essere distrutti.

5. Elementi facoltativi

Si definiscono come *facoltativi* i seguenti requisiti:

- Utilizzo della utility make per la compilazione
- Organizzazione dei vari file in cartelle (es. *./log*, *./bin*, *./src*, *./tmp*), e creazione delle stesse cartelle come parte del processo di compilazione
- Alla ricezione del messaggio EMERGENZA, riavvio dei processi PFC1, PFC2, PFC3 anziché arrestare il sistema. La lettura del file *G18.txt* dovrà riprendere dal punto in cui è stata interrotta.

- Utilizzo di macro per inizializzare le probabilità nel Generatore Fallimenti.

6. Regole per la presentazione del progetto.

Il progetto assegnato è valido fino alla sessione di Febbraio 2021 compresa.

Il progetto deve essere svolto in gruppi di massimo 3 persone.

E' necessario consegnare:

- Il **codice sviluppato** inclusi tutti i file necessari alla sua compilazione.
- Una **relazione** sul progetto. Ogni gruppo di lavoro dovrà produrre **una sola relazione**, i cui autori saranno quindi tutti i membri del gruppo stesso. La relazione dovrà essere in formato **pdf**. La relazione dovrà essere **massimo di 8 pagine**. Si consiglia di dare evidenza alle scelte progettuali relativi all'architettura complessiva del sistema, ed agli elementi facoltativi implementati.
- Un archivio **.zip** oppure **.tar.gz** contenente il codice e la relazione dovrà essere caricato sul sito del corso, seguendo l'apposito link che verrà reso disponibile per ciascun appello.
- *Non si accettano consegne in formati differenti da quelli sopra indicati*

Contenuto della relazione. La relazione dovrà contenere:

- Informazioni su tutti gli autori e data di consegna
 - o Nome, Cognome, Numero di matricola, Indirizzo e-mail
 - o Data di consegna
- Istruzioni dettagliate per compilazione ed esecuzione
- Sistema obiettivo: caratteristiche SW e HW, ad esempio la distribuzione Linux utilizzata
- Indicazione degli elementi facoltativi realizzati, riempiendo opportunamente la seguente tabella:

Elemento Facoltativo	Realizzato (SI/NO)	Metodo o file principale
Utilizzo Makefile per compilazione		
Organizzazione in folder, e creazione dei folder al momento della compilazione		
Riavvio di PFC1, PFC2, PFC3 alla ricezione di EMERGENZA		
Utilizzo macro nel Generatore Fallimenti per indicare le probabilità.		
PFC Disconnect Switch sblocca il processo se bloccato, e lo riavvia altrimenti.		
(continua da sopra) In entrambi i casi, il processo in questione deve riprendere a leggere dal punto giusto del file G18.txt.		

- Progettazione ed implementazione: presentazione ad alto livello della soluzione adottata, e principali caratteristiche o scelte progettuali
- Esecuzione: presentare e commentare una esecuzioni tipo del programma, fornendo evidenza del suo corretto funzionamento per i casi ritenuti più critici o interessanti.