Fly-By-Wire: LabSO edition

Andrea Ceccarelli
andrea.ceccarelli@unifi.it

Progetto per il corso di Sistemi Operativi
Anno Accademico 2019-2020

NOTA

La lettura di queste slides è fortemente raccomandata, ma HA SOLO SCOPO ESPLICATIVO del progetto.

La specifica del progetto è presente sul sito del corso.

(Pochissime) nozioni di background

Cosa è il Fly-By-Wire

Sistema che sostituisce i tradizionali comandi di volo diretti (cioè direttamente connessi agli elementi da controllare, meccanicamente o tramite un sistema idraulico) con un sistema di comando elettronico digitale

Esempio: operando cloche e manette, il comando è trasmesso tramite segnali elettrici a uno o più computer che elaborano e comandano attuatori (che agiscono sul sistema fisico)

https://it.wikipedia.org/wiki/Fly-by-wire

Boeing 777 (1998)

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. http://www2.coe.pku.edu.cn/tpic/20119263710178.pdf

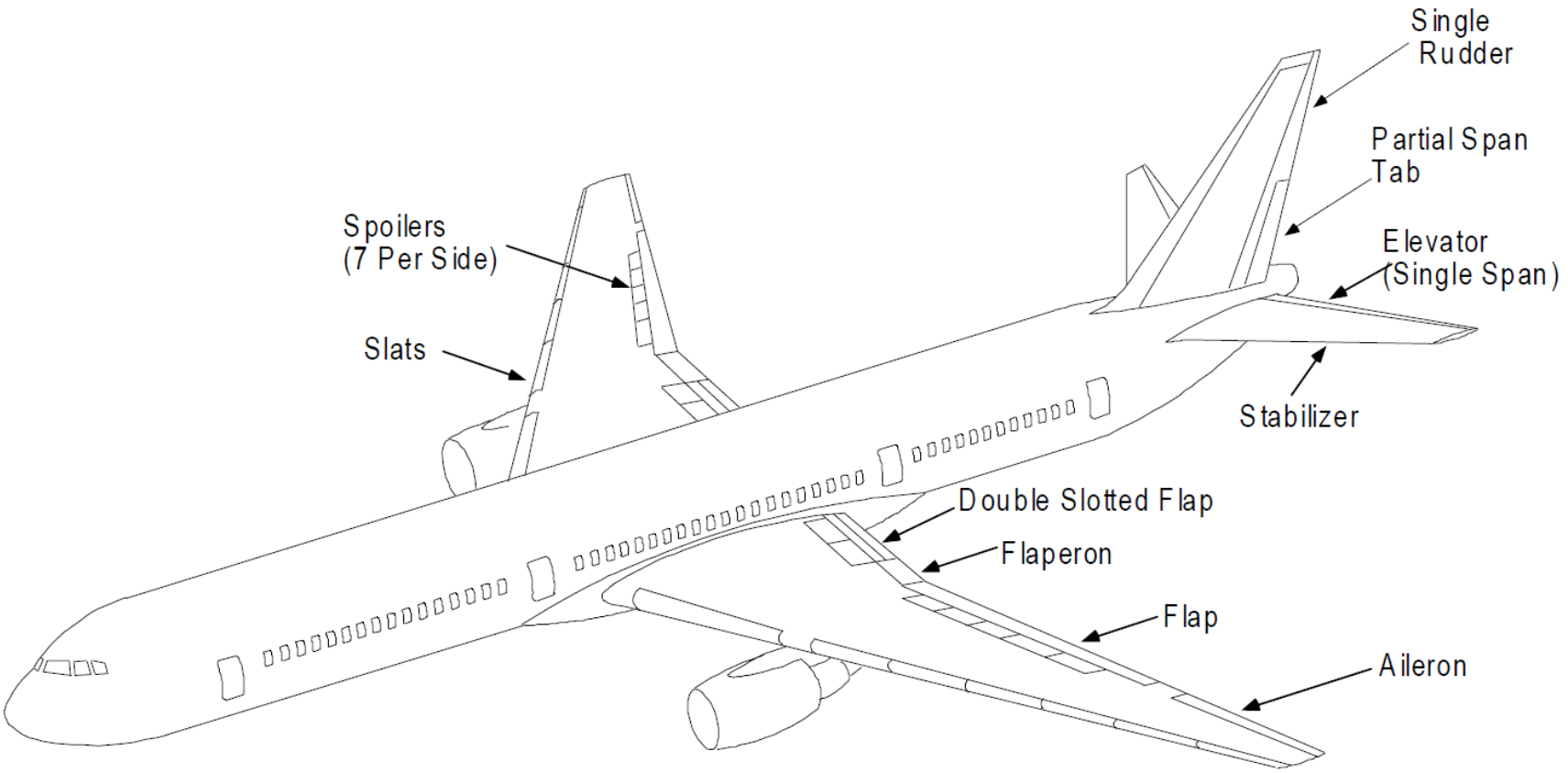


FIGURE 1 777 FLIGHT CONTROL SURFACES

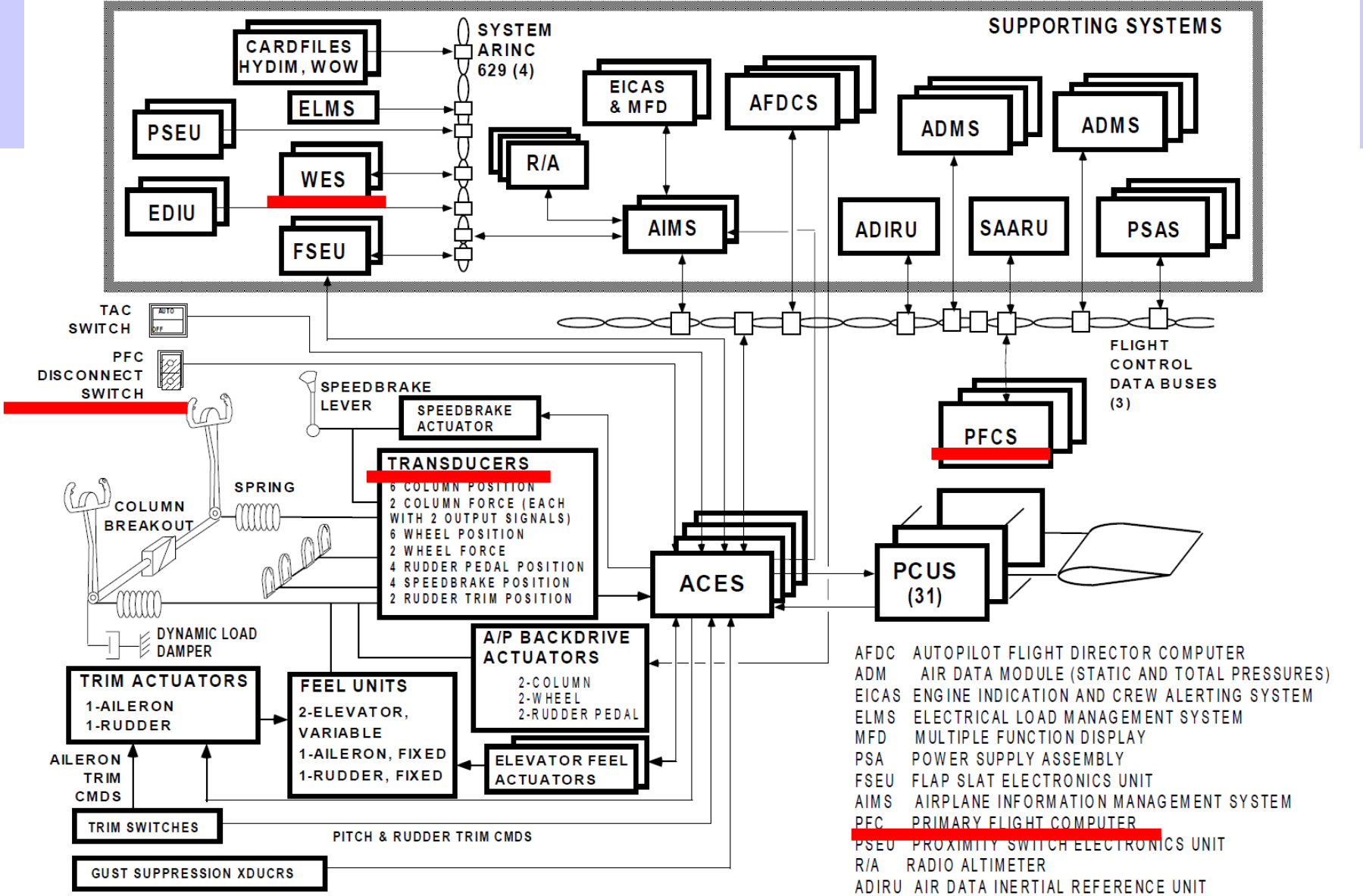


FIGURE 2 777 PRIMARY FLIGHT CONTROL SYSTEM OVERVIEW

Pochi elementi (e semplificati) utili per il progetto

Multiple redundant hardware

- Triple(-triple) redundant PFC architecture, triple channels with triple dissimilar lanes in each channel
- The PFC can be dispatched with one failed lane: maintenance alert is generated for maintenance attention. The PFC can also be dispatched with one failed channel: flight deck status message is generated requiring replacement of a PFC channel […].

NMEA 0183

Standard di comunicazione di dati utilizzato soprattutto in nautica e nella comunicazione di dati satellitari GPS

Nel formato NMEA:

- $GPGGL indica la posizione geografica

Esempio:

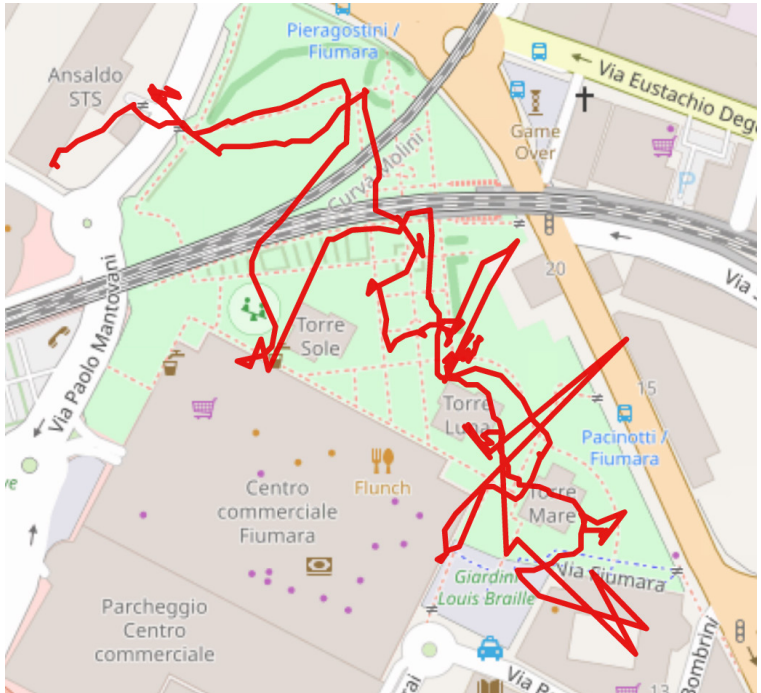
- $GPGLL,4424.8422,N,00852.8469,E,122359,V*35
  - Latitude 44 deg. 24.8422 min North
  - Longitude 8 deg. 52.8496 East
  - Fix taken at 12:23:59 UTC

http://aprs.gids.nl/nmea/

Dati NMEA usati per il progetto

Corrispondono ai dati raccolti da un GARMIN G18 in ambiente aperto:

- Possibili «salti» dovuti alla imprecisione del dispositivo e alle condizioni operative
- https://www.gpsvisualizer.com/

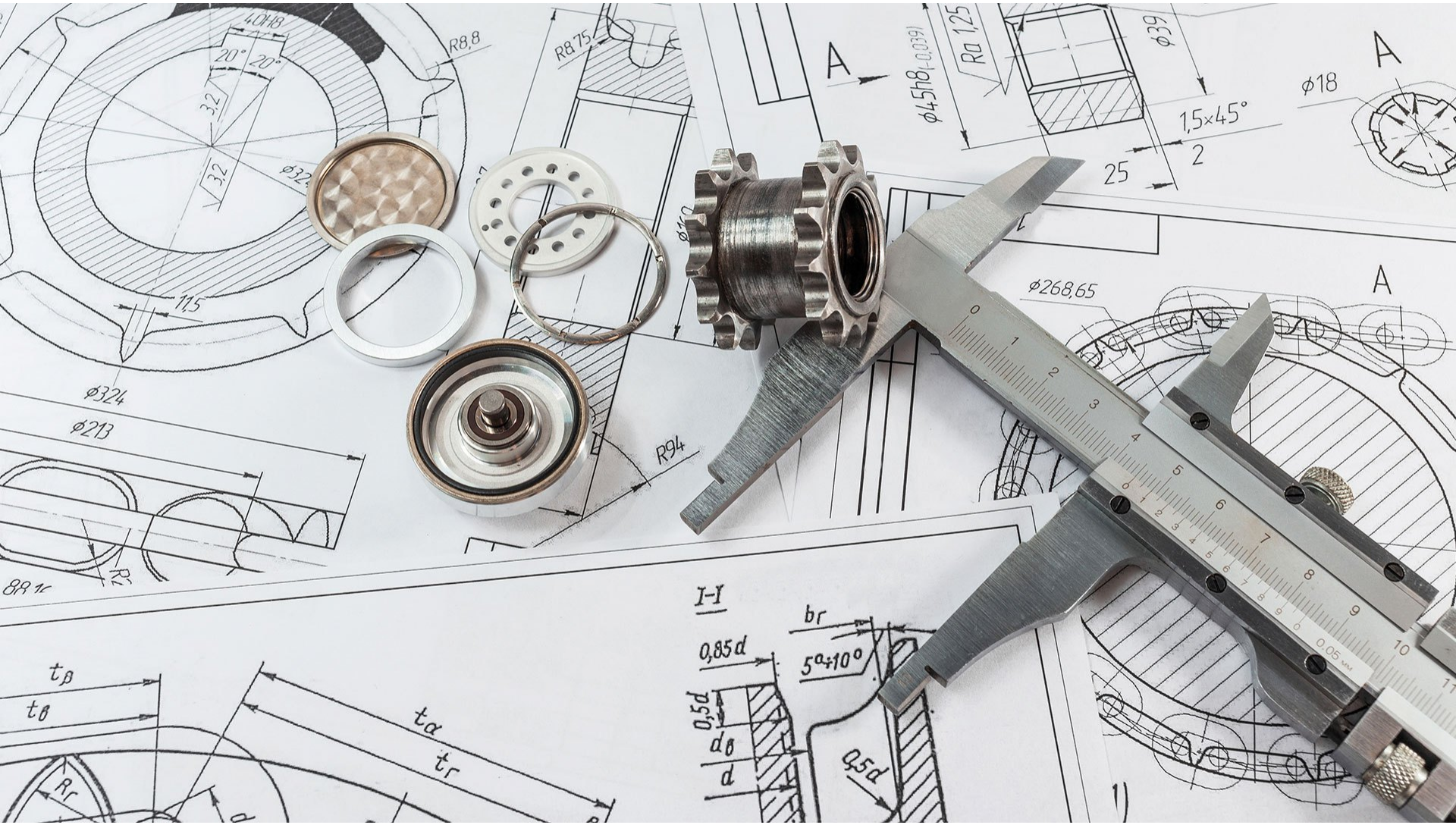


Come convertire coppie di coordinate NMEA in una distanza

Ci sono moltissimi riferimenti, ad esempio:

- https://stackoverflow.com/questions/365826/calculate-distance-between-2-gps-coordinates
- Alcune osservazioni sul codice al link di sopra (che non è per C):
  - La math.h include le funzioni sin, cos, arctan2, la macro M_PI, etc.
  - Il raggio della terra va calcolato rispetto alla latitudine, ovvero https://rechneronline.de/earth-radius/
  - Meglio operare in metri (raggio della Terra in metri), viste le distanze ridotte per il codice NMEA in input
  - GPGLL vi fornisce un valore come 4424.8422, ma molti algoritmi vogliono una sintassi quale 44.248422

Presentazione del progetto



Obiettivo

Obiettivo del progetto è costruire una architettura, ovviamente **estremamente stilizzata, rivisitata e difforme**, di un sistema Fly-By-Wire.

Il seguito delle slides entrano nei dettagli del progetto richiesto.

Richieste implementative

Se non diversamente specificato, le seguenti richieste implementative sono da intendersi prescrittive, equivalenti alla parola MUST secondo l'RFC 2119.

MUST This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.

Il sistema richiesto

I componenti considerati sono:

- PFC
- Transducers
- Generatore Fallimenti
- WES
- PFC Disconnect Switch

Ciascun componente architetturale di sopra è rappresentato da **almeno un processo, secondo quando descritto di seguito**.

PFC

Il PFC è triplicato, ovvero è composto da tre processi identici, che effettuano la **stessa operazione** in parallelo.

Chiameremo questi tre processi PFC1, PFC2, PFC3.

PFC

Il comportamento funzionale di PFC1, PFC2, e PFC3 è il seguente. Ad ogni secondo, ciascun PFC1, PFC2, PFC3

- Acquisisce una nuova stringa NMEA *GPGLL* dal file *G18.txt*, e da questa estrae le coordinate
- Calcola la distanza percorsa, rispetto alle coordinate acquisite all'istante precedente, e conseguentemente calcola la velocità attuale (all'avvio, la velocità è 0)
- Comunica la velocità elaborata a Transducers, tramite le seguenti modalità:
  - PFC1 comunica tramite una socket
  - PFC2 comunica tramite una pipe
  - PFC3 comunica tramite scrittura su un file condiviso

Generatore Fallimenti

Il generatore fallimenti è un processo che agisce sui processi PFC1, PFC2, o PFC3 nel seguente modo:

- Ad ogni istante di tempo, seleziona in modo casuale uno tra PFC1, PFC2 o PFC3, e:
  - 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 successivo calcolo della velocità, effettuando un left shift di 2 bits della velocità calcolata (dopo arrontondamento –e cast-- a intero).
- Effettua 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)

(ovviamente PFC1, PFC2, PFC3 devono essere in grado di ricevere i segnali sopra indicati)

Transducers

Transducers è un (o più) processo che:

- Acquisisce ad ogni istante la velocità inviata da PFC1, PFC2, PFC3
- Effettua il log rispettivamente nei file *speedPFC1.log*, *speedPFC2.log*, *speedPFC3.log*

(A causa dell'azione del Generatore Fallimenti, PFC1, PFC2, o PFC3 potrebbero non inviare il dato. In questo caso, semplicemente Transducer non riceverà il valore dal processo in questione.)

WES

Il WES è un processo che controlla continuamente lo stato di PFC1, PFC2, PFC3, e notifica eventuali problemi.

- Ad ogni istante, accede ai valori registrati su speedPFC1.log, speedPFC2.log, speedPFC3.log
- Se sono concordi, segnala un messaggio di OK
- Se due su tre processi sono concordi, e un processo è discorde
  - Invia un messaggio di ERRORE al PFC Disconnect Switch, indicando il processo discorde
- Se i tre valori sono discordi:
  - Invia al PFC Disconnect Switch un messaggio di EMERGENZA

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

PFC Disconnect Switch

Quando riceve dal WES un messaggio di ERRORE:

- Controlla lo stato del processo. Alcuni approcci:
  - `kill(PID, 0)` vi dice se il processo esiste (a meno di race conditions…).
  - `fscanf` su `/proc/pid_N/status`
- (opzionale) e lo «aggiusta» (sblocca, o riavvia). Approcci:
  - `signal, exec`
- (opzionale) Il processo deve ripartire a leggere dal punto giusto del file G18.txt.
  - Qualunque soluzione è valida. Esempio: tenere il numero dell'ultima riga letta da chiunque tra i tre processi in un file separato
- Registra le proprie attività nel file switch.log.

Quando riceve dal WES un messaggio di EMERGENZA:

- Termina l'applicazione.

Modalità di avvio e terminazione

Il programma deve essere avviato da una singola shell.

Si deve poter indicare il percorso del file G18.txt come parametro dell'eseguibile, ad esempio

```
$a.out /home/ceccarelli/labso/G18.txt
```

L'esecuzione termina quando l'ultima stringa GPGLL del file G18.txt è elaborata dalla PFC, o a causa di EMERGENZA. Al termine dell'esecuzione, è raccomandato distruggere tutti i processi creati.

Elementi facoltativi

E' considerato elemento facoltativo (e premiale)

- Utilizzo Makefile per compilazione
- Strutturazione dei file in folder (es. ./log, ./bin, ./src, ./tmp), e creazione degli stessi come parte del processo di compilazione
- Riavvio dei tre PFC1, PFC2, PFC3 al momento della ricezione del messaggio EMERGENZA, invece di arresto del sistema
- Inizializzazione delle probabilità del generatore fallimenti tramite macro.

Q&A Time


GRACIAS
THANK
YOU
ARIGATO
SHUKURIA
BOLZÏN
MERCI
BÏYAN
SHUKRIA
TASHAKKUR ATU
SUKSAMA
MEHRBANI
GRAZIE
DANKSCHEEN
JUSPÁXAR


Q
&
A