

# RELAZIONE PROGETTO SISTEMI OPERATIVI (2022/2023)

Zanolli Stefano(VR456008), Oualid Maftah(VR456187), Valeria Stuardi(VR422276)

## 1. File del progetto:

- Makefile
- F4Server.c
- F4Server\_functions.h
- F4Server\_functions.c
- F4Client.c
- F4Client\_functions.h
- F4Client\_functions.c

-----

## 2. Compilare ed eseguire :

(nel terminale, compila i file + rimuove le IPC utilizzate)

- make all

- ./F4Server dim1 dim2 simbolo1 simbolo2 (sul primo terminale)

- ./F4Client nome\_utente\_1 (sul secondo terminale)

- ./F4Client nome\_utente\_2 (sul terzo terminale)

### 3. Scelte progettuali:

- Per la connessione tra Server e i Client abbiamo usato un set di SEMAFORI
- Per la comunicazione tra Server e Client abbiamo usato una MsgQueue
- No limite max delle dimensioni delle matrice
- Per scelta progettuale ammettiamo che ci può essere lo stesso nome su più “giocatori” perchè si lavora sui PID dei Client
- $SERVER = 0$ ,  $CLIENT = 1$
- Come richiesto da progetto vengono controllati tutti gli argomenti passati nei parametri
- Per lavorare con la matrice, importante è la struttura matrix\_info
- Creo handle\_signals che ha il compito di selezione dei segnali specifici e ignorare tutti gli altri e dare a questi segnali, un handler “sinalg\_handler”
- Signal\_handler gestisce caso per caso la ricezione di ogni segnale non ignorato precedentemente
- Inizializzo il set di semafori (mask: 0777 -> libero) e li inizializzo
- Inizializzo la MsgQueue (mask: 0666 -> libero), vado a creare la Shared memory e faccio l'attach
- La funzione Blocca è un decremento al semaforo, quella Sblocca lo incrementa
- Per il client, creo un playerInfo che contiene Nome e PID e mtype.
- SIGUSR1 gestisce vittoria o pareggio
- SIGUSR2 gestisce il messaggio di vincita dopo l'abbandono
- SIGINT gestisce l'abbandono effettivo
- SIGTERM gestisce quando il server termina (chiuso)

