

## Hospital OS

Il progetto per un corretto funzionamento necessita dei seguenti file:

- ProcessGenerator.c
- Triage.c
- Paziente.c
- Reparto.c
- Dottore.c
- Configurazione.ini
- Sintomi.txt
- File header corrispondenti e necessari ad ogni file
- Makefile

### Specifiche dei comandi make:

1. Il comando **make** da terminale servirà per la compilazione dei vari file .c sopra elencati
2. Il comando **make run** avvierà il progetto lanciando l'esecuzione di ProcessGenerator.c con stampe esclusive a video.
3. Il comando **make file** avvierà il progetto lanciando l'esecuzione di ProcessGenerator.c con scrittura esclusiva su file.
4. Il comando **make both** avvierà il progetto lanciando l'esecuzione di ProcessGenerator.c con stampa a video e scrittura su file.

### **ProcessGenerator.c**

ProcessGenerator crea il semaforo che gestisce il numero di pazienti che posso accedere all'ospedale e lo inizializza al numero che viene impostato nel file Configurazione.ini;

Crea un semaforo binario e lo inizializza a 0 per sincronizzare il triage e i processi pazienti per evitare che quest'ultimi possano provare l'accesso alla coda messaggi del triage prima che questa venga creata dal triage, dopo di che viene lanciato Triage.c;

Dopo di che entra in un loop che termina solo quando riceve il segnale SIGQUIT che viene utilizzato per la generazione continua di processi pazienti.

### **Triage.c**

Triage crea il numero di reparti sulla base della quantità che viene letta dal file Configurazione.ini, dopo di che crea una FIFO per ogni reparto creato in precedenza;

Crea una coda di messaggi su cui ogni paziente scriverà un messaggio del tipo

"Sintomo;PIDpaziente" e incrementa il semaforo binario paziente-triage creato da

Processgenerator.c così da permettere ai pazienti di poter accedere alla coda di messaggi del triage;

Legge dalla coda di messaggi il sintomo e il PID di ogni paziente, attribuisce una priorità al sintomo del paziente appena letto dalla coda e scrive sulla FIFO di un reparto scelto tra quelli creati un messaggio del tipo "Sintomo;PIDpaziente;Priorità".

### ***Paziente.c***

Paziente cerca di accedere al semaforo creato da Processgenerator che gestisce il numero di pazienti che posso accedere all'ospedale e quando riesce lo decrementa di 1;

Sceglie un sintomo a random tra quelli elencati nel file Sintomi.txt;

Si collega alla coda creata da triage e scrive un messaggio del tipo "Sintomo;PIDPaziente" entra in un loop di attesa finchè non riceve il segnale SIGUSR1 che indica l'uscita del paziente dall'ospedale, quindi incrementa il semaforo di 1, oppure SIGQUIT, e termina.

### ***Reparto.c***

Reparto crea una coda di messaggi, lancia Dottore.c, dopo di che entra in un loop in cui legge dalla FIFO assegnatagli dal triage "Sintomo;PIDPaziente;Priorità" di ogni paziente assegnato a quel reparto e scrive sulla coda di messaggi del reparto "Sintomo;PIDPaziente" e assegnando a mtype la priorità.

### ***Dottore.c***

Dottore accede alla coda messaggi del Reparto ed entra in un loop in cui legge in alternanza, in base alla priorità o al FIFO, "Sintomo;PIDPaziente" e attraverso mtype la priorità;

Trascorso un tempo tra 2 e 5 sec. invia il segnale SIGUSR1 al paziente così che possa terminare.

- Tutti i processi se ricevono il segnale SIGQUIT terminano e nel caso in cui abbiano allocato delle strutture disallocano tutte quelle create prima di terminare.