

## Sistemi operativi – laboratorio

### Esercitazione 6: fifo

27 Aprile 2023 – 4 Maggio 2023

#### Ese\_1:

Scrivere un'applicazione Client-Server basata su FIFO. Il server crea una FIFO, e attende un messaggio. Il messaggio è un vettore di 2 interi  $[a, b]$ . Se  $a < b$ , il server stampa: "a è minore di b"; se  $a \geq b$  il server stampa: "a è maggiore o uguale a b". Dopo aver stampato a video la stringa, il server rimuove la FIFO, ed infine termina. Il client chiede all'utente due interi, invia i due numeri al server tramite la FIFO ed infine termina.

#### Ese\_2:

Estendere Ese\_1 affinché il server continui a leggere messaggi dalla FIFO creata. Il server rimuove la FIFO ed infine termina solo quando i due numeri ricevuti sono uguali, o dall'ultimo messaggio ricevuto sono passati più di 30 secondi.

#### Ese\_3:

Si vuole creare un applicativo Client-Server basato su FIFO. I messaggi inviati da un Client verso il Server sono istanze della struttura **Request**:

```
struct Request {  
    pid_t cPid;  
    int code;  
}
```

I messaggi inviati dal Server verso un Client sono istanze della struttura **Response**:

```
struct Response {  
    int result;  
}
```

Il processo Server esegue le seguenti operazioni:

1. Crea una FIFO di nome `fifo_server` dentro la directory `/tmp`
2. Legge da `fifo_server` un messaggio `req` di tipo `Request` (vedi sopra)
3. Crea un'istanza `resp` della struttura `Response` (vedi sopra) inizializzando il valore di `resp.result` uguale al quadrato di `req.code`.
4. Apre la FIFO `/tmp/fifo_client.cPid`, dove `cPid` è il valore della variabile `cPid` di `req` (se `req.cPid` è 123, il processo Server apre la FIFO `/tmp/fifo_client.123`).
5. Scrive `resp` in `/tmp/fifo_client.cPid`.
6. Ripete dal punto 2

Il processo Client esegue le seguenti operazioni:

1. Crea una FIFO di nome `fifo_client.pid` dentro `/tmp`, dove `pid` e' il PID del processo (se Client ha PID 123, il processo crea la FIFO `/tmp/fifo_client.123`)
2. Crea un'istanza `req` della struttura `Request` (vedi sopra), dove `cPid` contiene il PID del processo e `code` e' un numero generato casualmente (`rand()`).
3. Scrive `req` in `/tmp/fifo_client.cPid`.
4. Legge da `/tmp/fifo_client.cPid` un messaggio `resp` di tipo `Response` (vedi sopra).
5. Stampa a video il campo `result` di `resp`.
6. Cancella la FIFO `/tmp/fifo_client.cPid`, ed infine termina.

Se nessun messaggio e' ricevuto entro 30 secondi, o il segnale `SIGINT` viene inviato al processo Server, il quale cancella `fifo_server`, ed infine termina.