

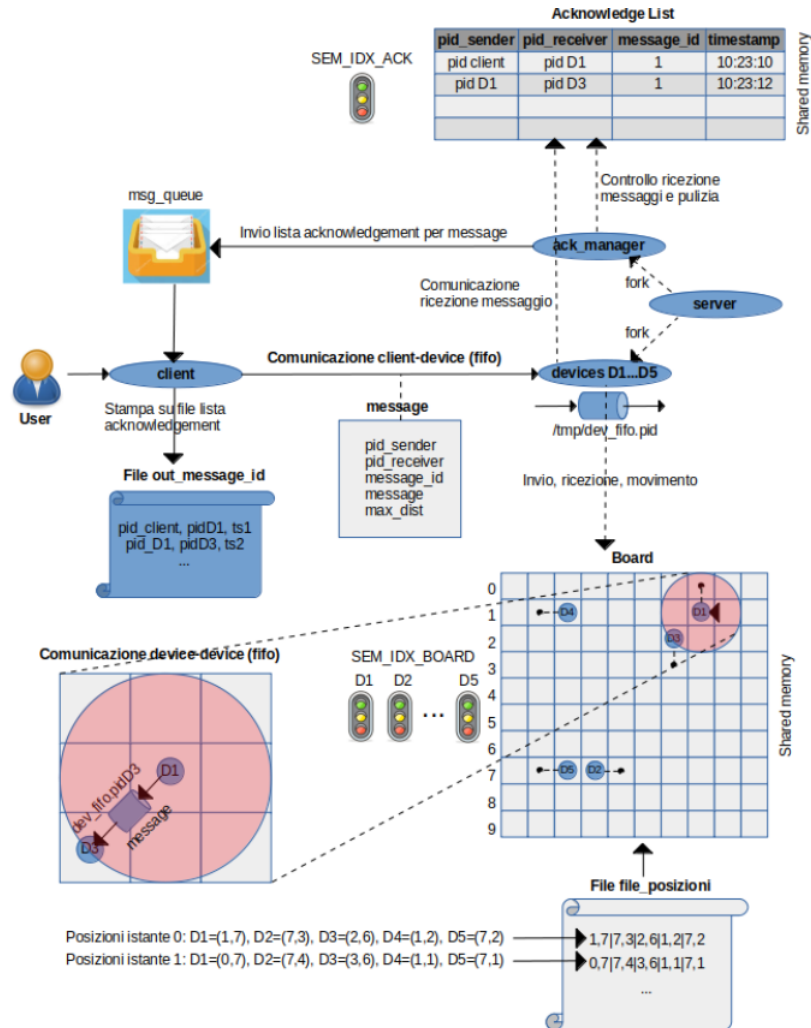
Progetto Sistemi Operativi

May 9, 2020

1 Introduzione

Trasferimento di messaggi tra dispositivi (2.1). I device si muovono all'interno di una scacchiera.

1.1 Schema funzionamento



2 Elementi

2.1 Device

5 Processi figli del processo server (2.2). Ognuno gestisce la propria FIFO (2.5.2). Invia i messaggi che ha agli altri device nell'area del suo raggio di comunicazione (2.1.2) e così anche gli altri. I dispositivi devono poter memorizzare e gestire più messaggi contemporaneamente.

2.1.1 Funzionamento

1. invio dei messaggi (se disponibili)
2. ricezione di messaggi
3. movimento

2.1.2 Device \rightarrow Device

Il nome della FiFO è *dev_fifo.pid*.

Messaggio La struttura del messaggio inviato è:

- *pid_sender*: pid del device
- *pid_receiver*: pid del del device ricevente
- *message_id*: id del messaggio
- *message*: stringa di testo.
- *max_dist*: numero positivo = raggio di invio del messaggio.

2.1.3 Acknowledgment List

Segmento di memoria condivisa generato da server 2.2. Gestisce il tracciamento di messaggi tra devices.

Messaggio La struttura del messaggio inviato è:

- *pid_sender*: pid del device
- *pid_receiver*: pid del del device ricevente
- *message_id*: id del messaggio
- *date_time*: data e ora di un passaggio

2.2 Server

Processo padre dei device 2.1 e di Ack_manager 2.3. Genera i segmenti di memoria relativi a acknowledge 2.1.3 e board 2.4. Crea i semafori per l'accesso ai segmenti di memoria in acknowledge, in board e al cambio posizone (movimento). Termina solo con SIGTERM (2.2.1). Scandisce il tempo dei movimenti.

2.2.1 SIGTERM

- Termina processi devices (2.1)
- Termina ack_manager (2.3)
- Termina coda di messaggi (2.3)
- Termina FIFO (2.1)
- termina memoria condivisa (2.4, 2.1.3)
- Termina semafori

2.3 Ack_manager

Processo figlio del processo Server2.2 Gestisce la lista condivisa di acknowledgement (2.1.3). Scandisce ad intervalli regolari di 5 secondi la lista per controllare se tutti i dispositivi hanno ricevuto il messaggio. In caso positivo invia subito la lista di acknowledgements al client (2.3.1). Ack_manager comunica con Client tramite coda di messaggi. Rimuove i messaggi dalla lista condivisa(2.1.3).

2.3.1 Ack_manager → Client

Il nome della coda di messaggi è *msg-queue*

2.4 Board (Scacchiera)

Scacchiera 10x10. Segmento di memoria condivisa generato da server 2.2. In posizione i,j ha scritto il PID del device (2.1) che è in quella posizione. Default cella = 0. I movimenti dei devices sulla scacchiera avvengono a turno ogni 2 secondi (tempo dato dal server). La sincronizzazione dei movimenti avviene tramite semaforo (2.4.2).

2.4.1 Posizioni

File posizione. Direttive di spostamento dei device.

2.4.2 Semaforo

Il semaforo si chiama *SEM_IDX_BOARD*.

2.5 Client

Processo generato dall'utente. Il client comunica con il Device tramite FIFO (2.5.2). Più client possono inviare messaggi contemporaneamente ai dispositivi. Quando riceve il messaggio da parte di acknowledgement (2.1.3), genera un file di nome *out_message.id.txt*(2.5.1) dove *message_id* è l'id del messaggio. Una volta generato il file il client termina.

2.5.1 Output file

Lista di 5 acknowledgement che identificano i passaggi fatti dal messaggio con i relativi istanti di tempo

2.5.2 Client → Device

Il nome della FIFO è *dev_fifo.pid*.

Messaggio La struttura del messaggio inviato è:

- *pid_sender*: pid del client
- *pid_receiver*: pid del del device ricevente
- *message_id*: id del messaggio

- *message*: stringa di testo.
- *max_idist*: numero positivo = raggio di invio del messaggio.