

Prova Lab II – Corso A

24 Marzo 2022

Si chiede di realizzare un programma C, denominato *shmpipe*, che implementa lo schema di comunicazione tra processi mostrato in Figura 1.

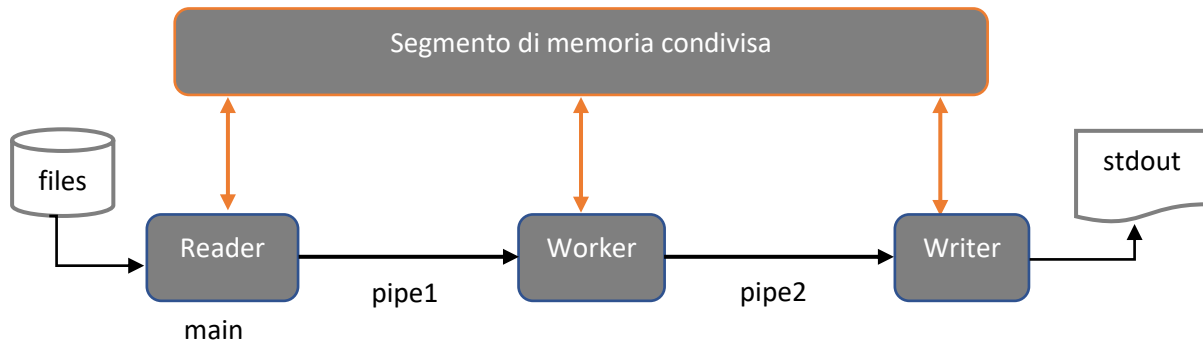


Figura 1 Architettura logica di connessione tra i processi Reader Worker e Writer

shmpipe prende come argomenti una lista di file binari contenenti numeri interi lunghi (long). Il processo *main* (in Figura 1 denominato *Reader*) crea un segmento di memoria condivisa di dimensione *MAX_FILE_SIZE*, e lancia 2 processi figli, uno chiamato *Worker* e l'altro *Writer*, connettendoli con le pipe senza nome *pipe1* e *pipe2* secondo lo schema logico mostrato nella Figura 1.

Il processo *Reader* legge gli argomenti passati alla funzione *main* (*argv*) uno alla volta, verificando che siano file regolari, e passa il contenuto di ogni file (uno alla volta) al processo *Worker* tramite il segmento di memoria condivisa. Il processo *Worker* si occuperà di fare un calcolo sugli elementi contenuti nel file e di inviare il risultato ottenuto al processo *Writer* tramite la *pipe2*. Il processo *Worker*, viene a conoscenza che nel segmento condiviso c'è un nuovo file da elaborare ogni volta che riceve un messaggio dalla *pipe1*, nella quale il processo *Reader* invia la size del file corrente copiato nel segmento condiviso (NOTA: lo studente, a sua discrezione, può anche passare la size del file corrente al *Worker* attraverso il segmento condiviso e notificare la presenza di un nuovo file da elaborare al *Worker* con un semplice messaggio di sincronizzazione nella *pipe1*).

Il processo *Writer* attende di ricevere un risultato dal *Worker* attraverso la pipe in ingresso *pipe2*, quindi stampa il risultato ottenuto sullo standard output.

Il calcolo effettuato dal *Worker* per ogni file ricevuto in ingresso tramite il segmento condiviso è il seguente:

$$result = \sum_{i=0}^{N-1} (i * file[i])$$

dove *N* è il numero di interi lunghi contenuti nel file e *result* è l'intero lungo che dovrà essere inviato al *Writer*. Ad esempio, supponendo che il file *fileX.dat* passato in input come argomento del *main* abbia dimensione 24 bytes, con il seguente contenuto (si ricorda che i *long* sono codificati con 8 bytes in sistemi Linux a 64bit):

3
2
4

il risultato calcolato dal *Worker* sarà: $N=3$, $result = \sum_{i=0}^{3-1} (i * file[i]) = (0 * 3 + 1 * 2 + 2 * 4) = 10$.

La sincronizzazione tra i processi *Reader* e *Writer* avviene attraverso un semaforo POSIX, e prevede che il processo *Reader* potrà scrivere un nuovo file nel segmento condiviso solo dopo che il processo *Writer* ha ricevuto il risultato del calcolo dal processo *Worker* sul file precedente. Il semaforo POSIX, a discrezione dello

studente, può essere istanziato all'interno del segmento condiviso (unnamed semaphore) oppure esternamente al segmento (named semaphore) -- in quest'ultimo caso, si fa notare che nella Figura 1 non si deve considerare la freccia tra il processo *Writer* ed il segmento condiviso.

Materiale fornito per il compito

Il materiale fornito dai docenti è il seguente:

- Testo del progetto (file *compito.pdf*)
- *Makefile* che è possibile modificare in base alle esigenze per compilare l'eseguibile *shmpipe*
- La directory *utils* dove è contenuto il file di *util.h* con alcune funzioni di utilità che è possibile utilizzare
- Un set di file di test (file1.dat, file2.dat, file3.dat, file4.dat) da utilizzare per verificare i risultati attesi

Risultato atteso

Configurando `MAX_FILE_SIZE` a 3MB, e lanciando il programma *shmpipe* con i file di test forniti, si dovrà ottenere un output simile a quello seguente:

```
> ./shmpipe file1.dat file2.dat file3.dat file4.dat  
  
3334678028  
569190  
12219904497  
4603140339274
```

Configurando `MAX_FILE_SIZE` ad 1KB, l'output atteso con i file di test forniti, è simile al seguente:

```
> ./shmpipe file1.dat file2.dat file3.dat file4.dat  
  
Il file file1.dat è troppo grande, il limite è 1024 byte  
Il file file3.dat è troppo grande, il limite è 1024 byte  
Il file file4.dat è troppo grande, il limite è 1024 byte  
569190
```

Consegna del compito

Il compito viene assegnato via il portale Classroom del corso. La consegna deve avvenire su Classroom. Tutti i file che implementano il compito (insieme ai file forniti dai docenti) devono essere consegnati in un unico file zip (o tgz) avente il seguente nome:

NomeCognome-Matricola.zip (o NomeCognome-Matricola.tgz).

I docenti verificheranno la funzionalità del compito consegnato compilando ed eseguendo il programma sulla macchina virtuale del corso (laboratorio2.di.unipi.it).