

Interprocess Communication in sistemi a memoria condivisa

Leonardo Bizzoni

October 23, 2023

La comunicazione tra processi basata sulla condivisione della memoria richiede che i **processi comunicanti allochino una zona di memoria condivisa**, solitamente nello spazio degli indirizzi del processo che la alloca.

Il tipo, la collocazione dei dati e la responsabilità di non scrivere nella stessa locazione simultaneamente è lasciata ai processi.

1 Implementazione

```
#include <sys/mman.h>
#include <unistd.h>

/* ... */

int fd = shm_open(name, flag, permission);
ftruncate(fd, size);
void *mem = mmap(0, size, mode, flag, fd, offset);

/* ... */

shm_unlink(name);

/* ... */
```

2 Esercizio

2.1 Sender

```
#include <stdio.h>
```

```

#include <stdlib.h>
#include <string.h>
#include <sys/mman.h>
#include <fcntl.h>
#include <unistd.h>

#define BUFF_SIZE 1024

int main(void) {
    char buff[BUFF_SIZE];

    int fd = shm_open("shm", O_CREAT | O_RDWR, 0640);
    ftruncate(fd, BUFF_SIZE);

    printf("L'id della shm è: %d\n", fd);

    char *shared_memory = (char *)mmap(0, BUFF_SIZE, PROT_WRITE, MAP_SHARED, fd, 0);

    printf("Cosa scrivo in shm?\n");
    read(0, buff, BUFF_SIZE);

    strcpy(shared_memory, buff);

    exit(EXIT_SUCCESS);
}

```

2.2 Receiver

```

#include <stdio.h>
#include <stdlib.h>
#include <sys/mman.h>
#include <fcntl.h>

#define BUFF_SIZE 1024

int main(void) {
    int fd = shm_open("shm", O_RDONLY, 0640);
    printf("L'id della shm è: %d\n", fd);

    if (fd != -1) {

```

```
char *shared_memory = (char *)mmap(0, BUFF_SIZE, PROT_READ, MAP_SHARED, fd, 0);

printf("Dati in shm: %s\n", shared_memory);
shm_unlink("shm");

exit(EXIT_SUCCESS);
}

printf("Non esiste il buffer di memoria condivisa \
associato alla chiave che hai fornito.\n");

exit(EXIT_FAILURE);
}
```