

Interprocess Communication

UNIX Shared Memory

Material from:

https://home.deib.polimi.it/fornacia/lib/exe/fetch.php?media=teaching:piatt_sw_rete_polimi:unix-shm.pdf

UNIX Shared Memory

- The parent and child processes are run in separate address spaces.
- A shared memory segment is a piece of memory that can be allocated and attached to an address space. Thus, processes that have this memory segment attached will have access to it.
- But, race conditions can occur!

UNIX Shared Memory

Procedure for Using Shared Memory

- Find a key. Unix uses this key for identifying shared memory segments.
- Use `shmget()` to allocate a shared memory.
- Use `shmat()` to attach a shared memory to an address space.
- Use `shmdt()` to detach a shared memory from an address space.
- Use `shmctl()` to deallocate a shared memory.

UNIX Shared Memory

Keys

- **Do it yourself:**

```
key_t SomeKey;  
SomeKey = 1234;
```

- **Ask the system to provide a private key using IPC_PRIVATE.**
- **Use ftok() to generate one for you:**

```
key_t = ftok(char *path, int ID);
```

Keys are global entities. If other processes know your key, they can access your shared memory.

UNIX Shared Memory

Asking for Shared Memory

- Use `shmget()` to request a shared memory: `shm_id = shmget(
 key_t key, /* identity key */
 int size, /* memory size */
 int flag); /* creation or use */`
- `shmget()` returns a shared memory ID.
- The flag, for our purpose, is either `0666 (rw)` or `IPC_CREAT|0666`.
- Include the following:

```
#include <sys/types.h>  
#include <sys/ipc.h>  
#include <sys/shm.h>
```

UNIX Shared Memory

Attaching Shared Memory

- Use `shmat()` to attach an existing shared memory to an address space:

```
shm_ptr = shmat(  
    int shm_id, /* ID from shmget() */  
    char *ptr, /* use NULL here */  
    int flag); /* use 0 here */
```

- `shmat()` returns a void pointer to the memory.

UNIX Shared Memory

Detaching and Removing Shared Memory

- To detach a shared memory, use `shmdt(shm_ptr);`
`shm_ptr` is the pointer returned by `shmat()`.
- After a shared memory is detached, it is still there.
You can re-attach and use it again.
- To remove a shared memory, use
`shmctl(shm_ID, IPC_RMID, NULL);`
`shm_ID` is the shared memory ID returned by `shmget()`. After a shared memory is removed, it no longer exists.