

Chia sẻ bộ nhớ giữa các process

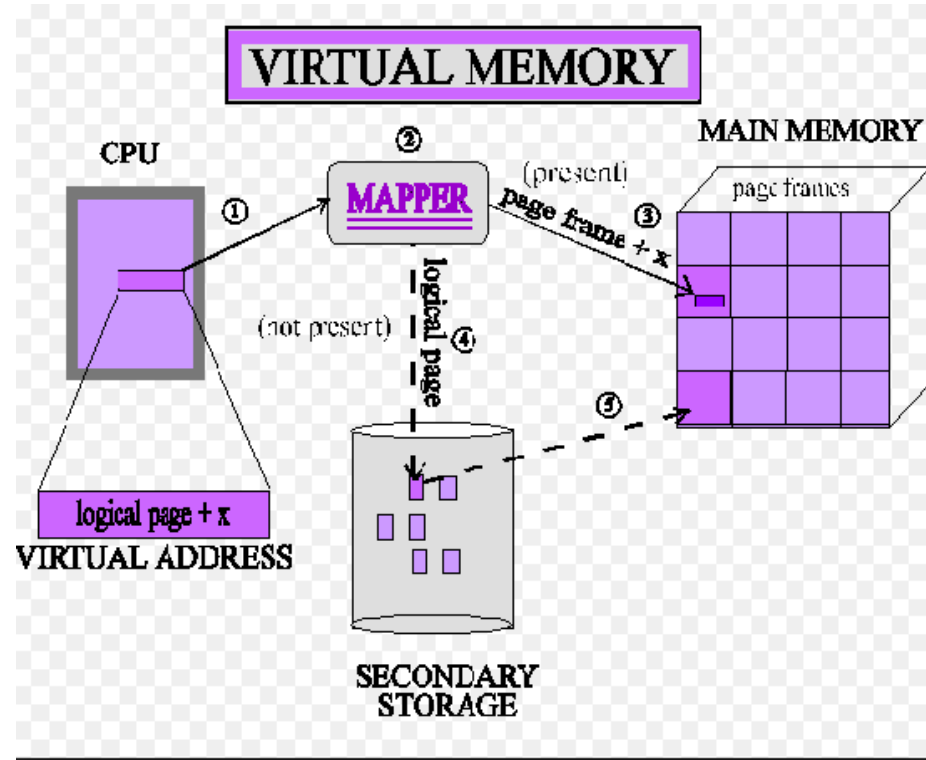
Giảng Viên: Lưu An Phú



- Memory page structure
- Shm API
- MMAP API
- Practice

memory page structure

- Struct page
 - `include/linux/mm_types.h`
- Permission
- Flag

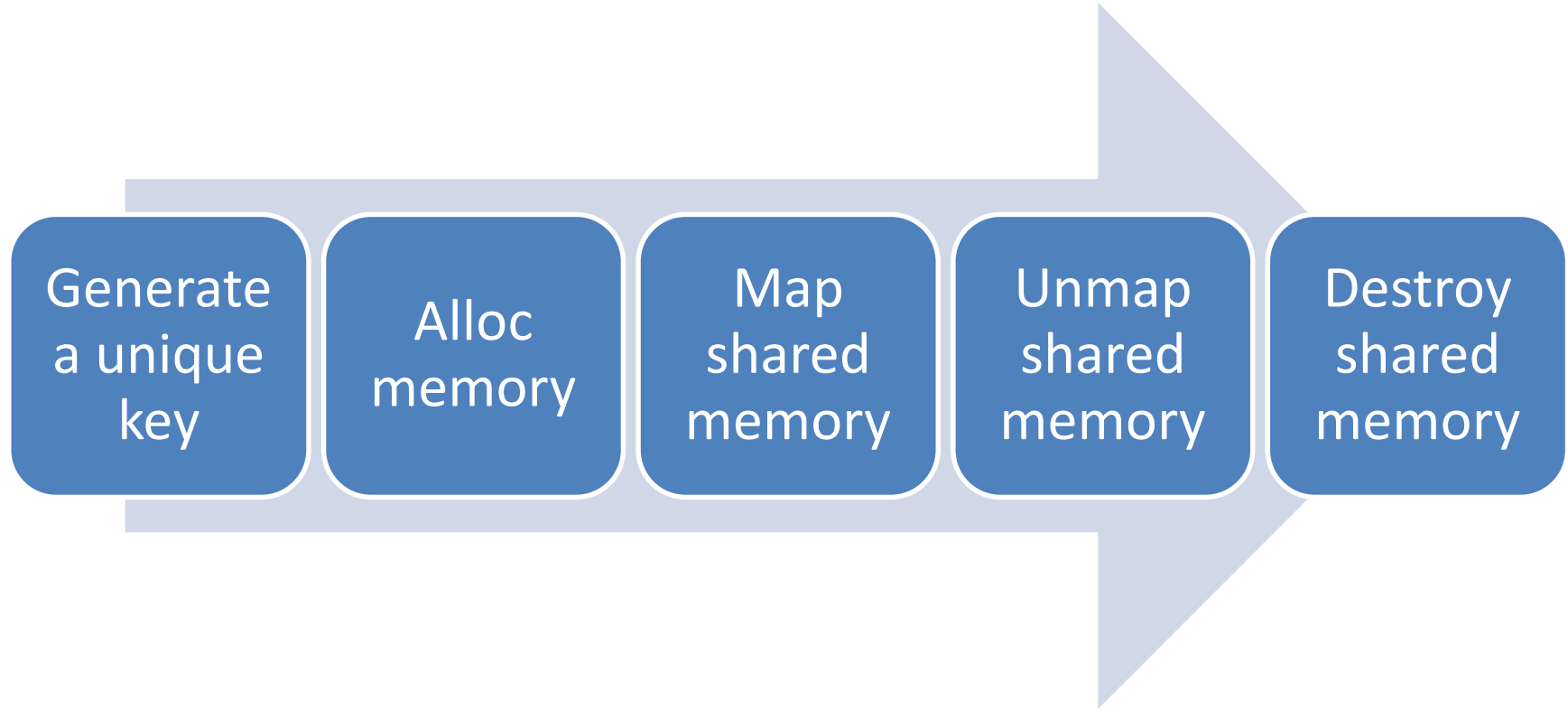


- shm API
 - Old API, support for all Unix like system
- Mmap
 - New API, some Unix like system doesn't support.
 - Can use to write in file.



shm API

Sequence diagram to share memory via shm API



generate a unique key from string

- `#include <sys/types.h>`
- `#include <sys/ipc.h>`
- `key_t ftok(const char *pathname, int proj_id);`
 - convert a pathname and a project identifier to a key

Alloc memory from key

- `int shmget(key_t key, size_t size, int shmflg);`
 - Allocate and return memory address for share memory
- `IPC_CREAT`
- `PROT_READ | PROT_WRITE`
- Permission like a file

- `void *shmat(int shmid, const void *shmaddr, int shmflg);`
- `shmaddr` should be `NULL`
- `SHM_RDONLY`

- `int shmdt(const void *shmaddr);`

Destroy shared memory

- `int shmctl(int shmid, int cmd, struct shmid_ds *buf);`
- `IPC_RMID`
 - Destroy shared mem after used.

Example code

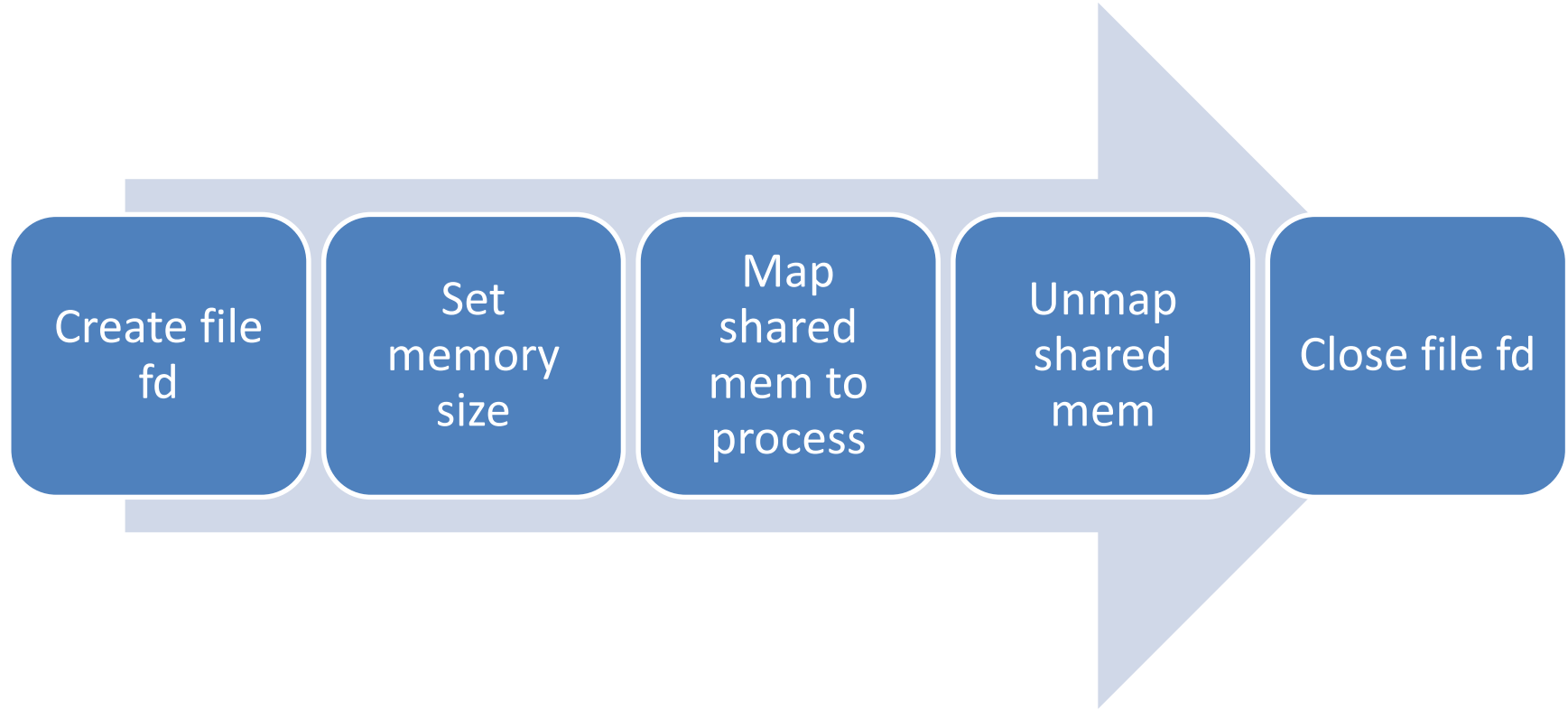
```
1  #include <iostream>
2  #include <sys/ipc.h>
3  #include <sys/shm.h>
4  #include <stdio.h>
5
6  int main()
7  {
8      ...// ftok to generate unique key
9      ...key_t key = ftok("shmfile",65);
10     ...
11     ...// shmget returns an identifier in shmid
12     ...int shmid = shmget(key,1024,0666|IPC_CREAT);
13     ...
14     ...// shmat to attach to shared memory
15     ...char *str = (char*) shmat(shmid,(void*)0,0);
16     ...
17     ...cout<<"Write Data : ";
18     ...gets(str);
19     ...
20     ...printf("Data written in memory: %s\n",str);
21     ...
22     ...//detach from shared memory
23     ...shmdt(str);
24     ...
25     ...return 0;
26 }
```

- Viết lại ứng dụng chat client server chạy trên cùng 1 máy tính. Sử dụng shared memory để giao tiếp với nhau.



mmap API

Sequence diagram to share memory via mmap API



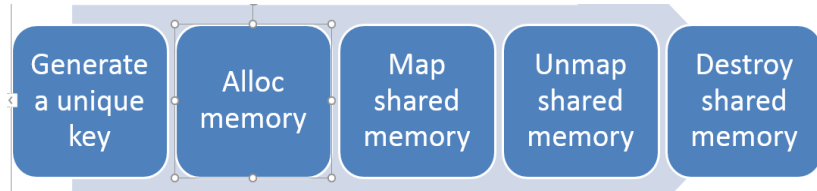
- `#include <sys/mman.h>`
- `void *mmap(void *addr, size_t length, int prot, int flags, int fd, off_t offset);`
 - `MAP_ANONYMOUS`
- `int munmap(void *addr, size_t length);`
- `shm_open(name, O_CREAT | O_RDWR, 0666);`
- `ftruncate(shm_fd, SIZE);`

Example code

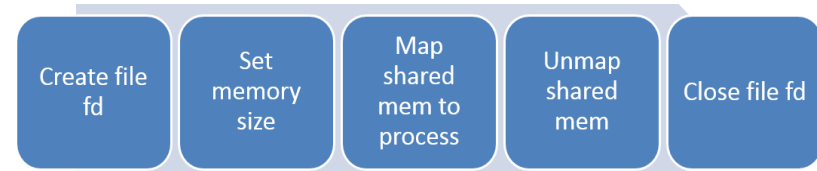
```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <fcntl.h>
5  #include <sys/shm.h>
6  #include <sys/stat.h>
7  ..
8  int main()
9  {
10     ...const int SIZE = 4096;
11     ...const char* name = "OS";
12     ...const char* message_0 = "Hello";
13     ...const char* message_1 = "World!";
14     ...int shm_fd;
15     ...void* ptr;
16
17     ...shm_fd = shm_open(name, O_CREAT | O_RDWR, 0666);
18     ...ftruncate(shm_fd, SIZE);
19     ...ptr = mmap(0, SIZE, PROT_WRITE, MAP_SHARED, shm_fd, 0);
20     ...sprintf(ptr, "%s", message_0);
21     ...ptr += strlen(message_0);
22     ...sprintf(ptr, "%s", message_1);
23     ...ptr += strlen(message_1);
24     ...return 0;
25 }
```

Compare shm and mmap API

Shm api



Mmap api



- Viết ứng dụng chat client – server trên cùng một máy tính. Sử dụng mmap để giao tiếp. Các đoạn chat sau khi thoát ứng dụng sẽ được lưu lại trên file.
 - Gợi ý: Nên define cố định độ dài mỗi lần ghi vào shared memory.

- Viết chương trình giả lập mô hình client server. Giữa 2 chương trình chia sẻ chung nhau 1 vùng nhớ (vùng nhớ data được tạo ra bởi server). Data được define theo struct sau:

```
Struct data {  
    char name[25];  
    int tuoi;  
}
```

- Đa số thời gian, server sẽ ngủ. Client sẽ yêu cầu user nhập thông tin tên, tuổi. Sau khi nhập xong client sẽ gửi signal để wake up server dậy. Server khi tỉnh dậy sẽ lấy thông tin từ share memory và ghi vào log file.
- Server và client được viết trên 2 file C riêng, không sử dụng fork.

Thank you

