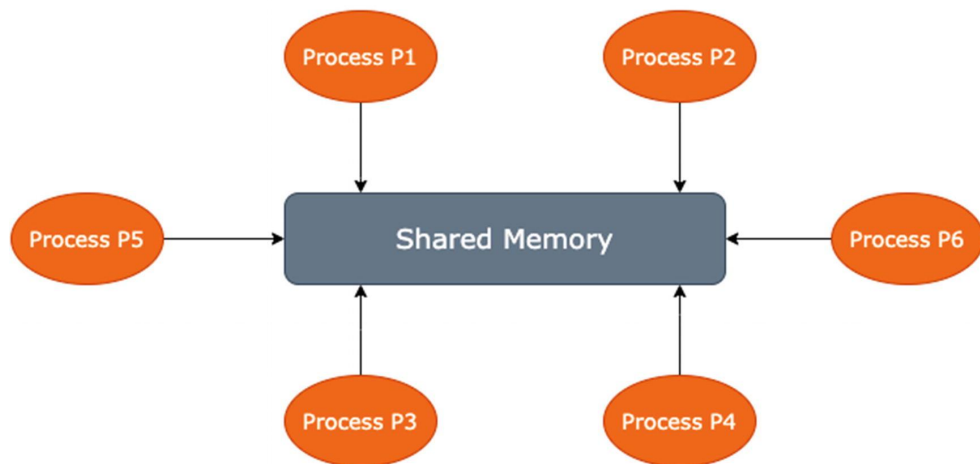


حافظه مشترک در XV6

نیما مدرس گرجی - غزل مینایی

حافظه مشترک - Shared Memory

- دسترسی توسط چند پردازش
- برای اشتراک گذاری داده‌ها
- استفاده بهینه از حافظه



پیاده سازی

جدول حافظه مشترک - shm_table:

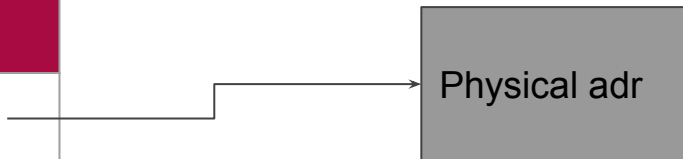
Mode	USR_R
	USR_RW
	برای حذف

● جدولی از سگمنت‌های حافظه مشترک (shmid_ds)

● نگهداری از اطلاعات مربوط به حافظه مشترک

● قفل گذاری

perm_info		ref_count	attached_processes	frame
id	mode			
7	USR_R	2	2, 3	
9	USR_RW	2	2, 4	



فراخوانی‌های سیستمی:

- `int sys_shm_getat(int id)`
- `int sys_shm_detach(int id)`
- `int sys_shm_ctl(int shmid, int cmd, struct shmid_ds)`

توابع کاربردی - `kalloc()`:

`char* kalloc(void)`

- برای اختصاص یک صفحه ۴۰۹۶ بایتی که سمت کرنل قابل استفاده باشد، از این تابع می‌توانید استفاده کنید.

توابع کاربردی - mappages():

```
static int mappages(pde_t *pgdir, void *va, uint size, uint pa, int perm)
```

- pgdir: process page table
- va: virtual address
- pa: physical address
- Create a pte: page table entry

```
// Page table/directory entry flags.  
#define PTE_P      0x001    // Present  
#define PTE_W      0x002    // Writeable  
#define PTE_U      0x004    // User  
#define PTE_PS     0x080    // Page Size
```

- برای ساختن نگاشت آدرس مجازی به فیزیکی از این تابع می‌توانید استفاده کنید.

توابع کاربردی - walkpgdir():

```
static pte_t * walkpgdir(pde_t *pgdir, const void *va, int alloc)
```

- برای پیدا کردن page table entry متناظر با آدرس مجازی (va) می‌توانید از این تابع کمک بگیرید.

توابع کاربردی - allocvm():

```
int allocvm(pde_t *pgdir, uint oldsz, uint newsz)
```

- برای اختصاص صفحه فیزیکی و نگاشت آن به جدول صفحه پرداز، میتوانید از پیاده‌سازی تابع allocvm() کمک بگیرید.

توابع کاربردی - lcr3():

```
static void lcr3(uint val)
```

- هر جا که جدول صفحه یک پردازنده را تغییر دادید، باید سخت افزار را از این تغییر آگاه کنید. برای این منظور، باید رجیستر CR3 را با جدول صفحه جدید پردازنده بروزرسانی کنید. آدرس فیزیکی page directory را به عنوان ورودی استفاده کنید.

ماکروی کاربردی - V2P():

```
#define V2P(a) (((uint) (a)) - KERNBASE)
```

- تبدیل آدرس مجازی به فیزیکی

ماکروی کاربردی - PTE_ADDR():

```
#define PTE_ADDR(pte) ((uint)(pte) & ~0xFFF)
```

- تبدیل pte به آدرس فیزیکی

int sys_shm_getat(int id)

- بررسی shm_table
- افزایش ref_count
- تغییر attached_processes
- تخصیص یک صفحه فیزیکی جدید با kalloc در صورت لزوم
- نگاشت بین حافظه مجازی و فیزیکی

int sys_shm_detach(int id)

- بررسی shm_table
- کاهش ref_count
- تغییر attached_processes
- آزادسازی در صورت علامت داشتن و
صفر بودن ref_count

```
int sys_shm_ctl(  
int shmid,  
int cmd,  
struct shmid_ds *buf)
```

cmd	IPC_STAT
	IPC_SET
	IPC_RMID

فراخوانی سیستمی - sys_shm_ctl:

● دستور IPC_SET:

در این دستور mode مربوط به حافظه مشترک با آیدی shmid به mode درون buf تغییر پیدا میکند. نوع دسترسی پردازش به حافظه نیز باید تغییر کند. پس کل page table entry ها را بررسی کنید. برای پیدا کردن pte با استفاده از آدرس مجازی از walkpgdir می‌توانید استفاده کنید. برای عوض کردن دسترسی ها از عملگرهای bitwise و مقدار PTE_W می‌توانید استفاده کنید. در نهایت سخت افزار را با lcr3 از این تغییر آگاه کنید.

فراخوانی سیستمی - sys_shm_ctl:

● دستور IPC_STAT:

برای حافظه با آیدی shmid اطلاعات id و mode در buf قرار دهید.

فراخوانی سیستمی - sys_shm_ctl:

● دستور IPC_RMID:

حافظه با آیدی shmid را برای آزاد شدن علامت زده بنماید. پردازش دیگری نمی‌تواند به آن متصل شود.

