



دانشگاه صنعتی امیرکبیر



آزمایشگاه سیستم عامل

جلسه پنجم: نحوه برقراری
ارتباط بین دو پردازنده

مدرس: مینا یوسفنژاد

Shmget

این تابع برای ایجاد یک بخش حافظه مشترک جدید یا دسترسی به یک بخش حافظه موجود استفاده می‌شود.

```
int shmget(key_t key, size_t size, int shmflg);
```

- **key:** برای ایجاد یک قطعه حافظه جدید، مقدار `IPC_PRIVATE` به آن داده می‌شود، که در این صورت یک شناسه جدید برمی‌گرداند که توسط سایر فرآیندها قابل استفاده است.
 - **size:** اندازه حافظه مورد نیاز را مشخص می‌کند. اگر قطعه از قبل وجود داشته باشد، اندازه نباید از اندازه اولیه بزرگتر باشد.
 - **shmflg:** برای تنظیم مجوزهای دسترسی و گزینه‌های خاص استفاده می‌شود. به عنوان مثال، `S_IRUSR` دسترسی خواندن و نوشتن به کاربر می‌دهد. `S_IWUSR` دسترسی خواندن و نوشتن به کاربر می‌دهد.
- در صورت موفقیت، این تابع یک مقدار `integer` به عنوان شناسه (`ID`) برای حافظه مشترک برمی‌گرداند.

shmat

این تابع برای اتصال یک بخش حافظه مشترک به فضای آدرس فرآیندی که آن را فراخوانی می‌کند استفاده می‌شود، که به این فرآیند امکان دسترسی به حافظه مشترک را می‌دهد.

```
void *shmat(int id, const void *addr, int flags);
```

- **id:** شناسه حافظه مشترک که توسط **shmget** برگردانده شده است.
 - **addr:** آدرس مورد نظر برای اتصال به حافظه مشترک. اگر مقدار آن **NULL** باشد، سیستم به صورت خودکار یک آدرس انتخاب می‌کند.
 - **flags:** برای مشخص کردن گزینه‌های دسترسی به حافظه مشترک است؛ برای ساده‌سازی معمولاً مقدار ۰ به آن داده می‌شود.
- در صورت موفقیت، این تابع یک اشاره‌گر به محل اتصال حافظه مشترک برمی‌گرداند.

shmdt

این تابع برای جدا کردن یک بخش حافظه مشترک از فضای آدرس فرآیندی که به آن متصل است استفاده می‌شود.

```
int shmdt(const void *addr);
```

➤ **addr**: آدرس اشاره‌گر به قطعه حافظه مشترکی که باید از فضای آدرس جدا شود.

این تابع در صورت موفقیت مقدار ۰ برمی‌گرداند.

shmctl

این تابع برای انجام عملیات کنترلی روی بخش حافظه مشترک استفاده می‌شود. با استفاده از این تابع می‌توان حافظه مشترک را حذف یا مجوزهای دسترسی آن را تغییر داد.

```
int shmctl(int id, int cmd, struct shmid_ds *buf);
```

- **id**: شناسه حافظه مشترک.
- **cmd**: عملیات مورد نظر را مشخص می‌کند.
- **IPC_RMID**: این دستور حافظه مشترک و شناسه آن را از سیستم حذف می‌کند.
- **IPC_SET**: برای تغییر مالکیت یا قوانین دسترسی حافظه مشترک.
- **IPC_STAT**: اطلاعات حافظه مشترک را در ساختار داده‌ای **buf** ذخیره می‌کند.
- این تابع در صورت موفقیت مقدار ۰ برمی‌گرداند.

shmflg

S_IRUSR دسترسی خواندن برای صاحب (کاربر) این حافظه مشترک.

S_IWUSR دسترسی نوشتن برای صاحب (کاربر) این حافظه مشترک.

S_IRGRP دسترسی خواندن برای گروه کاربر.

S_IWGRP دسترسی نوشتن برای گروه کاربر.

S_IROTH دسترسی خواندن برای سایرین (کاربران دیگر).

S_IWOTH دسترسی نوشتن برای سایرین (کاربران دیگر).

S_: "Set" or "Standard"

I: "Individual" "Inode"

R or W: "Read" or "Write."

USR, GRP, OTH: USR for "User" (the owner), GRP for "Group,"

and OTH for "Others."

Control Flags

این پرچم‌ها رفتار ایجاد یا دسترسی به حافظه مشترک را کنترل می‌کنند.

IPC_CREAT: این پرچم مشخص می‌کند که اگر حافظه مشترکی با **key** مورد نظر وجود ندارد، باید یک حافظه جدید ایجاد شود.

IPC_EXCL: این پرچم تنها در صورتی به کار می‌رود که **IPC_CREAT** نیز تنظیم شده باشد. با این پرچم، اگر یک بخش حافظه مشترک با همان **key** از قبل وجود داشته باشد، خطا برگردانده خواهد شد. این پرچم برای اطمینان از ایجاد حافظه جدید استفاده می‌شود و از تداخل با حافظه‌های موجود جلوگیری می‌کند.

مثال: اگر بخواهیم فقط زمانی حافظه ایجاد شود که قبلاً حافظه‌ای با آن کلید وجود نداشته باشد، از ترکیب **IPC_CREAT | IPC_EXCL** استفاده می‌کنیم.

آزمایش

ارتباط بین دو فرآیند با استفاده از حافظه مشترک
برنامه‌ای بنویسید که:

- حافظه مشترکی ایجاد کند.
- فرآیند فرزند را ایجاد کند.
- فرآیند والد پیامی در حافظه مشترک بنویسد.
- فرآیند فرزند پیام را از حافظه مشترک بخواند و آن را چاپ کند.

تمرین ۱

تغییر دسترسی‌های حافظه مشترک

برنامه‌ای بنویسید که:

- حافظه مشترکی ایجاد کند.
- دسترسی آن را به خواندن فقط برای کاربر تغییر دهد.
- بررسی کند که آیا نوشتن در حافظه پس از تغییر دسترسی‌ها امکان‌پذیر است یا خیر.