

گزارش پروژه: اشتراک‌گذاری حافظه بین دو فرآیند با استفاده از fork در C

مقدمه

در این پروژه، دو فرآیند مجزا با استفاده از fork ایجاد می‌شود که از طریق حافظه اشتراکی (shared memory) با یکدیگر ارتباط برقرار می‌کنند.

موارد گزارش

1. fork اسکرین شات:

```
pid_t pid = fork();

if (pid == 0) {
    int *data = (int *)shm_ptr;

    while (1) {

        int num = *data;

        printf("عدد دریافت شده: %d\n", num);

        sleep(1);
    }
} else if (pid > 0) {
    int *data = (int *)shm_ptr;

    while (1) {

        int num = rand() % 100;

        *data = num;

        printf("عدد تولید شده: %d\n", num);

        sleep(1);
    }
} else {
    perror("fork failed");
    exit(1);
}
```

توضیح:

- در این خط کد، تابع fork فراخوانی می‌شود و مقدار بازگشتی آن در متغیر pid ذخیره می‌شود.
- اگر pid برابر با 0 باشد، فرآیند فعلی در حال اجرا فرآیند فرزند است.
 - اگر pid بزرگتر از 0 باشد، فرآیند فعلی فرآیند والد است و pid شناسه فرآیند فرزند است.
 - اگر pid منفی باشد، خطایی رخ داده است.

2. حافظه اشتراکی اسکرین شات:

```
10     key_t shm_key = 12345;
11
12
13     int shm_id = shmget(shm_key, SHM_SIZE, IPC_CREAT | 0666);
14     if (shm_id < 0) {
15         perror("shmget failed");
16         exit(1);
17     }
18
19
20     void *shm_ptr = shmat(shm_id, NULL, 0);
21     if (shm_ptr == (void *)-1) {
22         perror("shmat failed");
23         exit(1);
24     }
```

توضیح:

در این کد:

- shm_key یک کلید عددی برای شناسایی حافظه اشتراکی است.
- SHM_SIZE اندازه حافظه اشتراکی به بایت است.
- IPC_CREAT به shmget می‌گوید که اگر حافظه اشتراکی با کلید مشخص شده وجود نداشته باشد، آن را ایجاد کند.
- 0666 مجوزهای دسترسی به حافظه اشتراکی را تنظیم می‌کند.
- shmget یک شناسه برای حافظه اشتراکی ایجاد می‌کند و آن را در متغیر shm_id ذخیره می‌کند.
- shmat حافظه اشتراکی را به فضای آدرس فرآیند فعلی متصل می‌کند و آدرس آن را در shm_ptr ذخیره می‌کند.

3. نحوه کار حافظه اشتراکی

حافظه اشتراکی توسط سیستم عامل مدیریت می‌شود و به هر فرآیندی که به آن متصل شده باشد، اجازه می‌دهد به طور همزمان به آن دسترسی داشته باشد.

فرآیندها می‌توانند از طریق آدرس حافظه اشتراکی که در shm_ptr ذخیره شده است، به داده‌ها در حافظه اشتراکی دسترسی داشته باشند.

4. ویژگی‌های fork

- fork یک کپی از فرآیند فعلی ایجاد می‌کند.
- فرآیند فرزند دارای کپی از فضای آدرس، فایل‌های باز و سایر منابع فرآیند والد است.
- fork یک روش کارآمد برای ایجاد فرآیندهای جدید است.
- fork می‌تواند برای ایجاد فرآیندهای والد-فرزند برای ارتباط بین فرآیندها استفاده شود.

5. ویژگی‌های حافظه اشتراکی

- حافظه اشتراکی به فرآیندها اجازه می‌دهد تا به طور کارآمد داده‌ها را به اشتراک بگذارند.
- حافظه اشتراکی سریعتر از سایر روش‌های اشتراک‌گذاری داده‌ها مانند پیام‌های بین فرآیندی (IPC) است.
- حافظه اشتراکی می‌تواند برای به اشتراک‌گذاری داده‌های بزرگ بین فرآیندها استفاده شود.