

DEADLOCK

بررسی ددلاک و نخ‌های POSIX

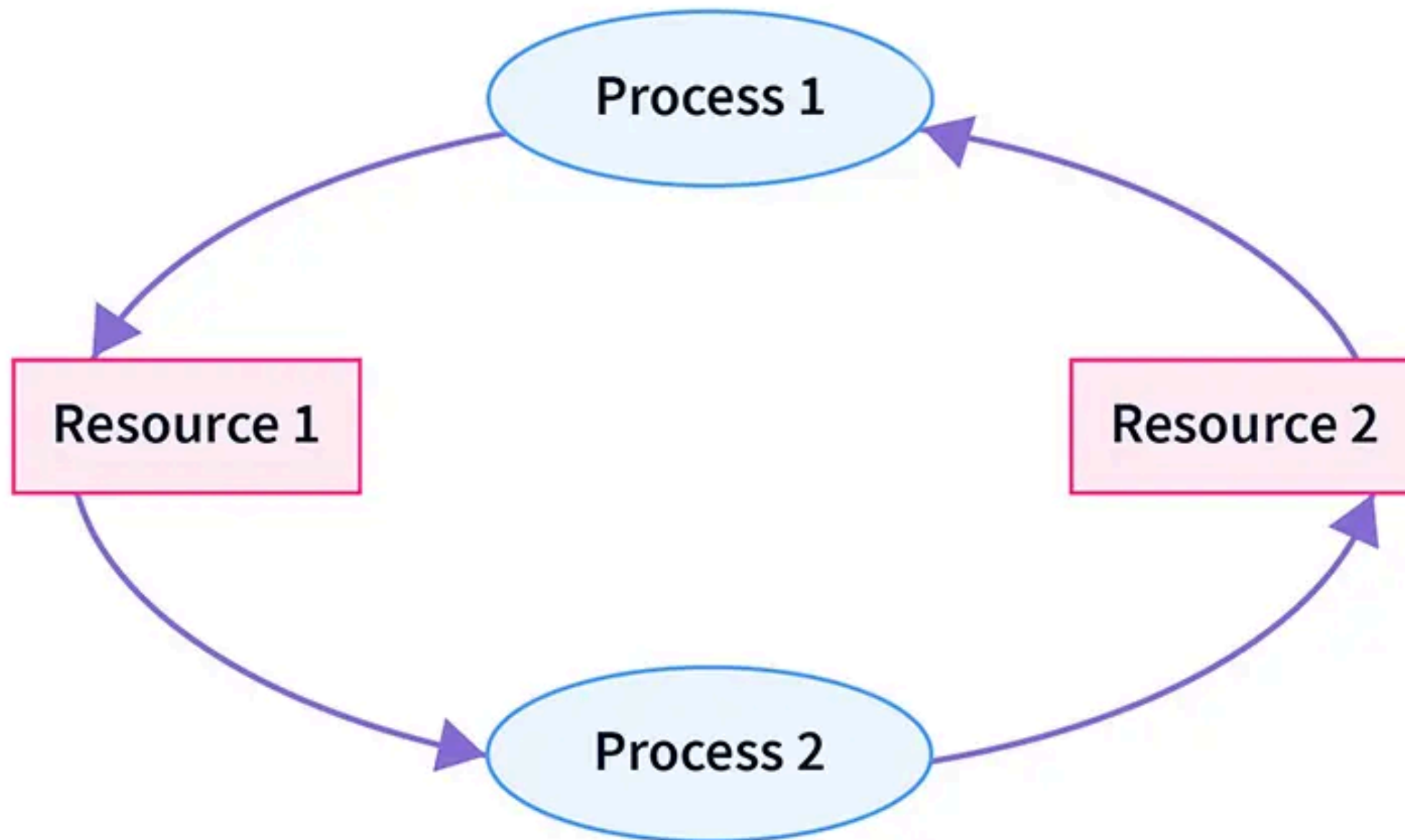
حسین تاتار - 40133014

پیش گزارش دهم آزمایشگاه سیستم عامل

1404/03/06

ددلاک چیست؟

- وضعیتی که در آن چند فرآیند منتظر منابعی هستند که توسط یکدیگر اشغال شده‌اند.
- منجر به توقف دائمی اجرای فرآیندها
- وابستگی چرخشی بین فرآیندها



شرایط لازم برای وقوع ددلاک

تخصیص انحصاری (Mutual Exclusion) 

در حال انتظار با حفظ منابع (Hold and Wait) 

عدم پیش‌گیری (No Preemption) 

انتظار چرخشی (Circular Wait) 



پیشگیری

مثلاً با تخصیص همه منابع در ابتدای اجرا

اجتناب

Banker's Algorithm - RAG

کشف و بازیابی

تشخیص چرخه، خاتمه دادن به فرآیند

نادیده گرفتن

Ostrich Algorithm

روش‌های مقابله با ددلاک





LINUX

آشنایی با نخه‌های POSIX



نخ‌های POSIX (Pthreads)

- استاندارد برای ایجاد و مدیریت نخ‌ها در یونیکس/لینوکس

-
- شامل توابعی مانند `pthread_create`, `pthread_join`, `pthread_mutex_lock`

-
- اجرا موازی کدها توسط چند نخ در یک فرآیند
- 

نحوه ایجاد نخ POSIX

ایجاد نخ POSIX با کد روبرو:

```
#include <pthread.h>


void* my_function(void* arg) {
    // عملیات نخ
    return NULL;
}

int main() {
    pthread_t t1;
    pthread_create(&t1, NULL, my_function, NULL);
    pthread_join(t1, NULL);
    return 0;
}
```



Race Condition

شرایط مسابقه و جلوگیری از آن



شرایط مسابقه (Race Condition)

- زمانی رخ می‌دهد که دو نخ به طور همزمان به داده مشترک دسترسی دارند.
- منجر به رفتار غیرقابل پیش‌بینی و خطاهای همزمانی می‌شود.

روش‌های جلوگیری

رعایت نظم دسترسی
به منابع مشترک

استفاده از Semaphore

استفاده از Mutex
(pthread_mutex_t)

استفاده از Mutex در POSIX

```
pthread_mutex_t lock;
```

```
pthread_mutex_init(&lock, NULL);
```

```
pthread_mutex_lock(&lock);
```

```
// کد بحرانی
```

```
pthread_mutex_unlock(&lock);
```

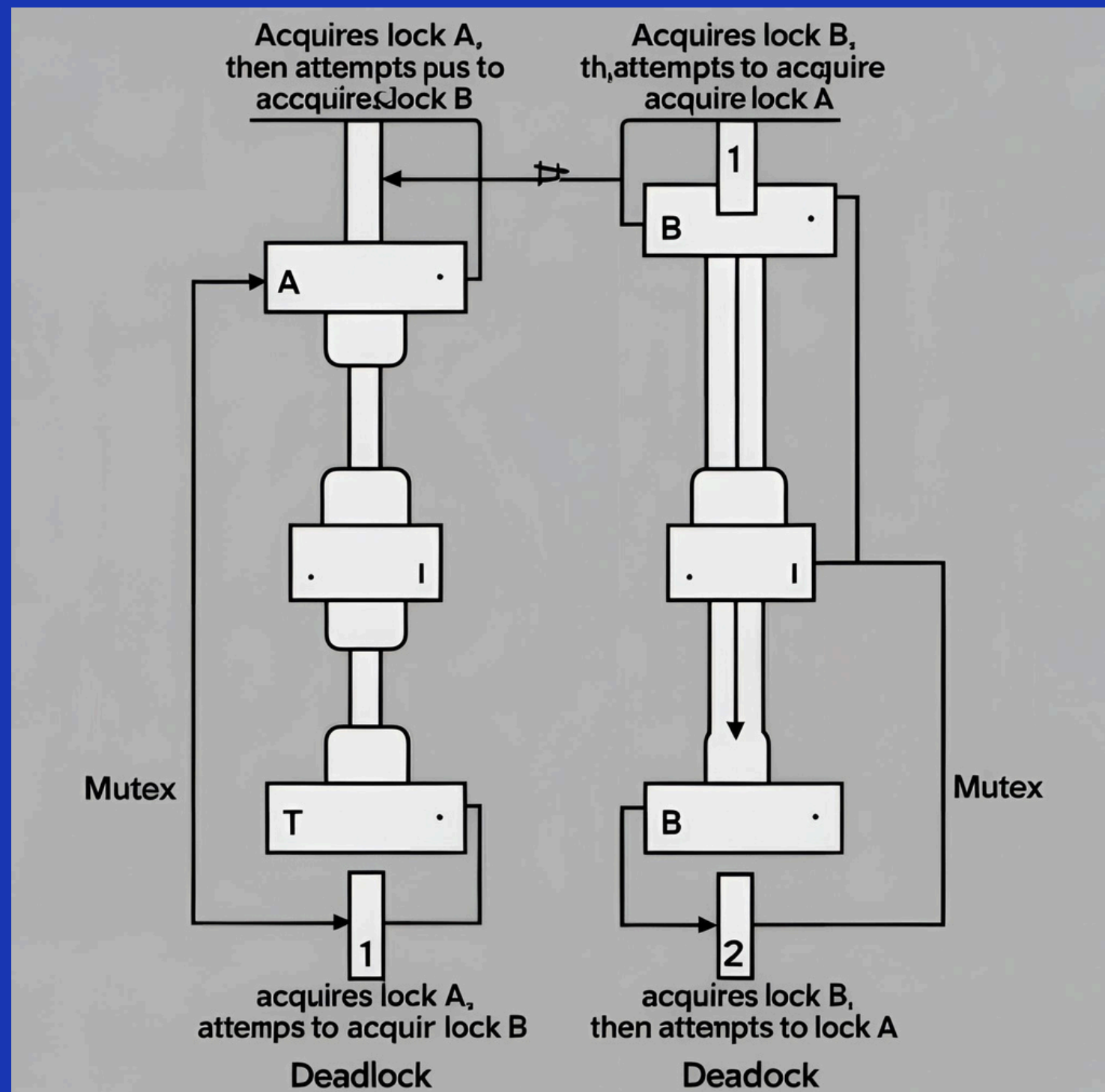
```
pthread_mutex_destroy(&lock);
```

کد نمونه کنترل دسترسی با Mutex:

مثالی از ددلاک

بین نخ ها

- نخ 1: گرفتن lock A، سپس تلاش برای lock B
- نخ 2: گرفتن lock B، سپس تلاش برای lock A
- نتیجه: هر دو نخ در انتظار یکدیگر میمانند.



ممنون از توجه شما
