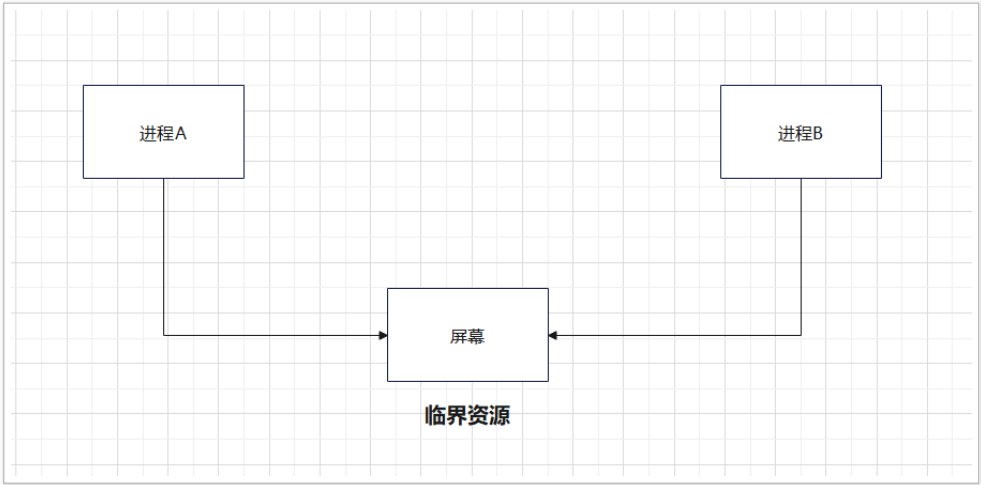


5.1 进程间通讯 - 信号量 (一)_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 5.1 进程间通讯 – 信号量 (一) 涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。

- 资源竞争：当多个进程在同时访问共享资源时，会产生资源竞争，最终最导致数据混乱
- 临界资源：不允许同时有多个进程访问的资源, 包括硬件资源 (CPU、内存、存储器以及其他外围设备) 与软件资源(共享代码段、共享数据结构)
- 临界区：访问临界资源代码



```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>
#include <sys/wait.h>

int main(void)
{
    pid_t cpid;

    cpid = fork();

    if (cpid == -1){
        perror("[ERROR] fork(): ");
        exit(EXIT_FAILURE);
    }else if (cpid == 0){
        while(1){
            printf("-----\n");
            printf("C Start.\n");
            sleep(1);
            printf("C End.\n");
            printf("-----\n");
        }
    }else if (cpid > 0){
        while(1){
            printf("-----\n");
```

```

        printf("P Start.\n");
        sleep(1);
        printf("P End.\n");
        printf("-----\n");

    }

    wait(NULL);

}

return 0;
}

```

- 互斥：同一时刻只有一个进程访问临界资源
- 同步：在互斥的基础上增加了进程对临界资源的访问顺序
- 进程主要的同步与互斥手段是 信号量
- 信号量: 由内核维护的整数, 其值被限制为大于或等于 0
- 信号可以执行如下操作:
 - 将信号量设置成一个具体的值
 - 在信号量当前值的基础上加上一个数值
 - 在信号量当前值的基础上减上一个数值
 - 等待信号量的值为 0
- 一般信号量分为 二值信号量 与 计数信号量
 - 二值信号量：一般指的是信号量 的值为 1, 可以理解为只对应一个资源
 - 计数信号量: 一般指的是值大于等于 2, 可以理解为对应多个资源
- 在 Linux 系统中查询信号量使用 `ipcs -s`

----- Semaphore Arrays -----

key semid owner perms nsems

函数头文件

```
#include <sys/types.h>
```

```
#include <sys/ipc.h>
```

```
#include <sys/sem.h>
```

函数原型

```
int semget(key_t key, int nsems, int semflg);
```

函数功能

创建一个信号量集合

函数参数

- key：由 `ftok()` 函数生成
- nsems：信号量的数量
- semflg：信号量集合的标志
 - `IPC_CREAT`：创建标志
 - `IPC_EXCL`：与 `IPC_CREAT` 标志一起使用, 如果信号量集合存在就报错
 - 权限标志

函数返回值

- 成功：返回信号量集合的 id
- 失败：-1, 并设置 errno

函数头文件

```
#include <sys/types.h>
```

```
#include <sys/ipc.h>
```

```
#include <sys/sem.h>
```

函数原型

```
int semctl(int semid, int semnum, int cmd, ...);
```

函数功能

信号集合控制函数，根据 cmd 决定当前函数的功能

函数参数

- semid：信号量集合的 id
- semnum：信号量的编号, 信号量的编号从 0 开始
- cmd：命令控制字
 - SETVAL: 设置信号量的值
 - GETVAL: 获取信号量的值
- ...：后面是属于可变参数列表, 根据不同的命令有不同的参数

函数返回值

- 成功：根据不同的命令有不同的返回值, 可以查看帮助文档关于 RETURN 的说明
 - GETNCNT the value of semncnt
 - GETPID the value of sempid
 - GETVAL the value of semval
 - GETZCNT the value of semzcnt.
- All other cmd values return 0 on success.

失败：返回 -1, 并设置 errno

1. 在使用命令时需要使用 union semun 共用体, 具体定义如下:
 2. union semun {
 3. int val; /* Value for SETVAL */
 4. struct semid_ds buf; / Buffer for IPC_STAT, IPC_SET */
 5. unsigned short array; / Array for GETALL, SETALL */
 6. struct seminfo __buf; / Buffer for IPC_INFO
 7. (Linux-specific) */
 8. };

示例

创建一个信号量集合, 集合中包含一个信号量, 并设置信号量的值为 1

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
```

```
#include <sys/ipc.h>
#include <sys/sem.h>

#define SEM_PATHNAME "."
#define SEM_PRO_ID 100

union semun{
    int val;
};

int main(void)
{
    int semid,ret;

    union semun s;

    key_t key = ftok(SEM_PATHNAME,SEM_PRO_ID);
    if (key == -1){
        perror("[ERROR] ftok() : ") ;
        return -1;
    }

    semid = semget(key,1,IPC_CREAT | 0666);
    if (semid == -1){
        perror("[ERROR] semget() : ") ;
        return -1;
    }

    s.val = 1;

    ret = semctl(semid,0,SETVAL,s);
    if (ret == -1){
        perror("[ERROR] semctl() : ") ;
        return -1;
    }

    return 0;
}
```

练习

请创建一个信号量集合，里面包含两个信号量，分别初始化为 0 和 1

全文完

本文由 简悦 SimpRead 优化，用以提升阅读体验

使用了 全新的简悦词法分析引擎 beta，点击查看详细说明

