

4.8 进程间通讯 - 共享内存 (一)_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 4.8 进程间通讯 – 共享内存（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。

- 共享内存是将分配的物理空间直接映射到进程的用户虚拟地址空间中，减少数据在内核空间缓存
- 共享内存是一种效率较高的进程间通讯的方式
- 在 Linux 系统中通过 ipcs -m 查看所有的共享内存

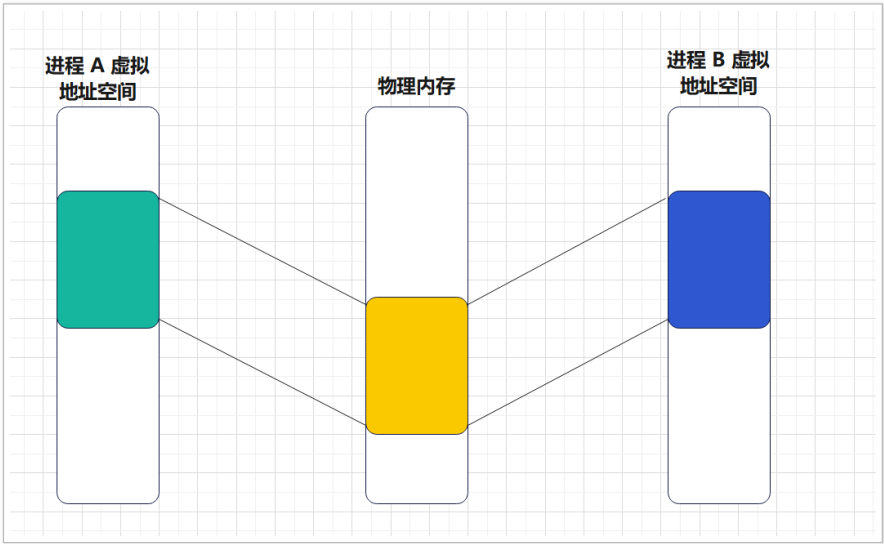
----- Shared Memory Segments -----

key shmid owner perms bytes nattch status

0x00000000 11 ben 600 524288 2 dest

0x00000000 16 ben 600 524288 2 dest

0x00000000 35 ben 600 4194304 2 dest



函数头文件

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

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

函数原型

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

函数功能

创建一个共享内存, 并返回 ID

函数参数

- key : 由 ftok() 函数返回
- size : 共享内存的大小
- shmflg : 共享内存标志

函数返回值

- 成功 : 返回 共享内存 id
- 失败 : 返回 -1, 并设置 errno

示例

创建一个共享内存, 并打印出共享内存 ID

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

#define PATHNAME "."
#define PRO_ID 100

#define SZ 256

int main(void)
{
    key_t key;
    int shmId, ret;

    key = ftok(PATHNAME, PRO_ID);
    if(key == -1){
        perror("[ERROR] key(): ");
        exit(EXIT_FAILURE);
    }

    shmId = shmget(key, SZ, IPC_CREAT | 0666);
    if(shmId == -1){
        perror("shmId(): ");
        exit(EXIT_FAILURE);
    }

    printf("shmId = %d\n", shmId);

    return 0;
}
```

函数头文件

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

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

函数原型

```
int shmctl(int shmId, int cmd, struct shmId_ds *buf);
```

1. 函数功能

1. 共享内存控制函数, 功能由具体的功能命令字决定

2. 函数参数

1. shmId : 共享内存 id
2. cmd : 控制命令字
 1. IPC_STAT : 获取 消息队列属性
 2. IPC_SET : 设置消息队列属性
 3. IPC_RMID : 删除消息队列属性, 用此命名时, 第三个参数为 NULL
3. buf : 共享内存属性结构体指针

3.

消息队列属性结构体定义如下:

```
struct ipc_perm {

    key_t __key; /* Key supplied to msgget(2) */

    uid_t uid; /* Effective UID of owner */

    gid_t gid; /* Effective GID of owner */

    uid_t cuid; /* Effective UID of creator */

    gid_t cgid; /* Effective GID of creator */

    unsigned short mode; /* Permissions */

    unsigned short __seq; /* Sequence number */

};
```

函数返回值

- 成功：返回 0，特殊命令字除外
- 失败：返回 -1

示例

- 创建一个共享内存后，输出共享内存 id, 删除共享内存

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

#define PATHNAME "."
#define PRO_ID 100

#define SZ 256

int main(void)
{
    key_t key;
    int shmid,ret;

    key = ftok(PATHNAME,PRO_ID);
    if(key == -1){
        perror("[ERROR] key(): ");
        exit(EXIT_FAILURE);
    }

    shmid = shmget(key,SZ,IPC_CREAT|0666);
    if(shmid == -1){
        perror("shmget(): ");
        exit(EXIT_FAILURE);
    }

    printf("shmid = %d\n",shmid);

    ret = shmctl(shmid,IPC_RMID,NULL);
    if (ret == -1){
        perror("[ERROR] shmctl(): ");
        exit(EXIT_FAILURE);
    }

    return 0;
}
```

练习

4.8 进程间通讯-共享内存(一)_物联网/嵌入式工程师-慕课网

创建共享内存, 并打印 id, 然后删除, 使用 ipcs -m 命令进行验证

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

