

## 4.9 进程间通讯 - 共享内存 (二)\_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 4.9 进程间通讯 – 共享内存 (二) 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。

### 函数头文件

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

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

### 函数原型

```
void *shmat(int shmid, const void *shmaddr, int shmflg);
```

### 函数功能

将进程地址空间映射到共享内存上

### 函数参数

- shmid : 共享内存 id
- shmaddr : 指定映射的到进程地址空间的起始地址
  - 指定为 NULL 时, 由系统选择映射的地址
- shmflg : 共享内存标志, 一般设置为 0

### 函数返回值

- 成功 : 返回映射到进程地址空间的起始地址
- 失败 : (void \*) -1, 并设置 errno

### 示例

创建共享内存, 并打印 id, 再删除共享内存, 使用 `ipcs -m` 命令测试

### 函数头文件

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

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

### 函数原型

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

### 函数功能

解除进程地址空间与共享内存的映射

### 函数参数

## 函数返回值

- 成功：返回 0
- 失败：返回 -1, 并设置 errno

## 示例

## 使用共享内存进行进程间通讯

```
#include <stdio.h>
#include <stdlib.h>
#include <string.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;
    void *addr = NULL;

    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);
    addr = shmat(shmId, NULL, 0);
    if (addr == (void *)-1) {
        perror("[ERROR] shmat(): ");
        return -1;
    }

    memset(addr, 'A', 10);

    shmdt(addr);

    return 0;
}
~
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.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, i;
    char buffer[10] = {0};
    void *addr = NULL;

    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);
addr = shmat(shmid, NULL, 0);
if (addr == (void *)-1){
    perror("[ERROR] shmat(): ");
    return -1;
}

memcpy(buffer,addr,10);

for (i = 0;i < 10;i++){
    printf(" %c ",buffer[i]);
}

putchar('\n');

shmdt(addr);

return 0;
}
```

### 练习

在两个进程使用共享内存之间传输文件

---

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

