

实验3——共享内存进程间通信实验

1.实验内容

利用Linux进程通信（IPC）的共享内存函数，实践操作系统的共享内存进程间通信的相关知识。共享内存函数由shmget、shmat、shmdt、shmctl四个函数组成。

- 任务一：

以实验2的进程创建为基础。创建一个子进程，在子进程中向共享内存中写入一个字符串。父进程等待子进程的结束，当子进程结束后，父进程读出共享内存中子进程写入的字符串。

- 任务二：

取消任务一中父进程对子进程的等待，再次编译，观察运行结果。

针对于本实验中任务一、二出现的结果的区别请在实验报告中予以解释和说明。

2.实验示例代码

- 需要用到的头文件

```
1 #include<sys/shm.h>
2 #include<sys/stat.h>
3 #include<unistd.h>
4 #include<sys/types.h>
5 #include<sys/wait.h>
```

- 初始化一个共享内存区域

```
1 int segment_id;
2 char *shared_memory;
3 const int size = 4096;
4
5 segment_id = shmget(IPC_PRIVATE, size, S_IRUSR | S_IWUSR);
6 shared_memory = (char*)shmat(segment_id, NULL, 0);
```

- 子进程中向共享内存区域写

```
1 shared_memory = (char*)shmat(segment_id, NULL, 0);
2 printf("Child write: Hi There!\n");
3 sprintf(shared_memory, "Hi There!");
```

- 父进程中从共享内存区域读

```
1 wait(NULL);
2 printf("Parent Read: %s\n", shared_memory);
3 shmdt(shared_memory);
4 shmctl(segment_id, IPC_RMID, NULL);
```

3.实验结果示例

```
wys@wys-VirtualBox:~/桌面/lab03$ ./shareMemory
Child Write: Hi There!
Parent Read: Hi There!
wys@wys-VirtualBox:~/桌面/lab03$ ./shareMemory
Child Write: Hi There!
Parent Read: Hi There!
wys@wys-VirtualBox:~/桌面/lab03$ ./shareMemory
Child Write: Hi There!
Parent Read: Hi There!
wys@wys-VirtualBox:~/桌面/lab03$ ./shareMemory
Child Write: Hi There!
Parent Read: Hi There!
```

有 wait(NULL)的情况

```
wys@wys-VirtualBox:~/桌面/lab03$ ./shareMemory_without_wait
Parent Read:
Child Write: Hi There!
wys@wys-VirtualBox:~/桌面/lab03$ ./shareMemory_without_wait
Parent Read:
Child Write: Hi There!
wys@wys-VirtualBox:~/桌面/lab03$ ./shareMemory_without_wait
Parent Read:
Child Write: Hi There!
wys@wys-VirtualBox:~/桌面/lab03$ ./shareMemory_without_wait
Parent Read:
Child Write: Hi There!
wys@wys-VirtualBox:~/桌面/lab03$ ./shareMemory_without_wait
Parent Read:
Child Write: Hi There!
wys@wys-VirtualBox:~/桌面/lab03$
```

没有 wait(NULL)的情况