



共享内存示例：

```
ptr = mmap(NULL, 1024, PROT_READ|PROT_WRITE, MAP_SHARED | MAP_ANONYMOUS, -1, 0);
//调用 mmap 函数创建一块匿名的共享内存区域。返回共享内存区域的起始地址。
//子进程向共享内存写入数据
if (0 == pid) {
    memcpy(ptr, "good morning", 13);
    munmap(ptr, 1024);
    exit(0);
}
//父进程等待子进程结束并读取共享内存数据
wait(NULL); -> puts(ptr); -> munmap(ptr, 1024);
//父进程等待子进程结束后，从共享内存中读取数据并输出，最后解除共享内存映射。
wait(NULL); --> puts(ptr); --> munmap(ptr, 1024);
```

资源竞争示例：

```
semid = semget(IPC_PRIVATE, 1, IPC_CREAT | IPC_EXCL | 0600);信号量创建
semctl(semid, 0, SETVAL, 1);用于对信号量集进行控制操作
文件初始化
fp = fopen(FILENAME, "w");以写入模式打开文件 /tmp/out，如果文件不存在则创建
fputs("0", fp); --> fclose(fp);
子进程调用该函数对文件进行读写操作
fp = fopen(FILENAME, "r+"); P(); // 取信号 fgetc(buf, BUFSIZE, fp);
rewind(fp); fprintf(fp, "%d", atoi(buf) + 1); fclose(fp); V(); // 还信号
static void V(P(void)
{ struct sembuf sop; sop.sem_num = 0; sop.sem_op = 1 | -1; sop.sem_flg = 0; }
```