

“扩展实验”选题

1.系统调用扩充（续）

在实验1的基础上，进一步实现如下的一组系统调用，并通过指定的测试用例。

系统调用mmap

```
#define __NR_mmap 92
long mmap(void *start, size_t len, int prot, int flags,
          int fd, off_t off);
```

- 功能：将文件或设备映射到内存中；
- 输入：
 - start: 映射起始位置
 - len: 长度
 - prot: 映射的内存保护方式，可取：PROT_EXEC, PROT_READ, PROT_WRITE, PROT_NONE
 - flags: 映射是否与其他进程共享的标志，可取：MAP_FILE, MAP_SHARED, MAP_PRIVATE
 - fd: 文件句柄
 - off: 文件偏移量
- 返回值：成功返回已映射区域的指针，失败返回-1;

系统调用munmap

```
#define __NR_munmap 93
int munmap(void * start, size_t len);
```

- 功能：将文件或设备取消映射到内存中；
- 输入：
 - start: 映射的指定地址
 - len: 区间长度
- 返回值：成功返回0，失败返回-1;

系统调用clone

```
#define __NR_clone 94
int clone(int (*fn)(void *), void *child_stack, int flags, void *arg);
```

- 功能：创建一个子进程；
- 输入：
 - fn: 子进程的主函数
 - child_stack: 指定新进程的栈，可为0。

- flags: 创建的标志, 如SIGCHLD、CLONE_VM, 其中CLONE_VM表示子进程共享父进程的地址空间、但使用不同的栈。
- arg: 主函数的参数
- 返回值: 成功则返回子进程的线程ID, 失败返回-1;

2.操作系统移植 (arm)

完善如下项目, 将Linux 0.11移植到arm平台。

- [linux_0.11_arm_imx6ull](#): 这是一个已有的移植项目, 可基于其中的v3分支进行开发。

3.操作系统移植 (risc-v)

在如下已有项目的基础上, 将Linux 0.11移植到risc-v平台

- [Linux_0.11_RISC-V_QEMU](#): 这是国防科大卢昶诚的移植, 可初步在Qemu上运行。
- [K210-Linux0.11](#): 这是聊城大学李志锐的移植, 目标是在K210开发板 (RISC-V平台) 上运行。
- [RISC-V Specifications](#): 这是RISC-V的原始文档, 重点参考其中的"Volume 1, Unprivileged Spec"和"Volume 2, Privileged Spec"。
- [RISC-V 硬件中文手册](#)

4.ELF可执行文件格式的支持

使内核支持运行ELF格式的可执行文件

5.独立页表的支持

使每个进程拥有一个自己的页表, 有独立的4G地址空间

6.FAT32文件系统的支持

使内核支持FAT32格式的文件系统

7.虚拟内存与交换分区的设计与实现

实现虚存和交换分区

8.内核级线程的设计与实现

实现内核级线程

9.鼠标驱动和简单的图形接口实现

实现简单的图形界面

10.网卡驱动与网络协议的设计与实现

实现简单的网络功能