

系统原有功能/特性

特性：保护模式，多终端。

原有功能：

- 五态进程,信号量
- 系统设置了 5 个终端，左上角为终端编号，1 ~ 5，每个进程在自己所属的终端进行键盘输入和屏幕输出.其中终端1只用于于特权级为1的系统任务,终端2~5用于特权级为3用户程序
- 开机后会自动运行 2 个系统任务（特权级为 1），分别为终端任务 **tty** 和控制台任务 **shell**. **tty** 任务实现多终端控制,**shell** 任务用于实现系统命令的输入和执行
- 用户程序可以占据终端 2 ~ 5，每个终端只能运行 1 个用户程序,但该用户程序可以在该终端利用 **fork()** 创建多个子进程.用户程序运行时会先初始化其所在终端，清除之前的用户程序的显示痕迹.用户进程的数量上限为20.
- **shell** 提供两种指令，指令输入以回车结束
 - **run**:运行指定编号的用户程序，若用户程序不到 4 个且该程序没有运行，则该程序将开始运行
 - **end**:结束指定编号的用户程序，若该用户程序正在运行，则该程序将结束运行
- 快捷键：
 - **shift + up** :将屏幕向上滑动，每次滑动 1 行，若已在顶端，则无反应
 - **shift + down** :将屏幕向下滑动，每次滑动 1 行，若已在底端，则无反应
 - **alt + F1** : 切换到终端1
 - **alt + F2** : 切换到终端2
 - **alt + F3** : 切换到终端3
 - **alt + F4** : 切换到终端4
 - **alt + F5** : 切换到终端5
 - 备注：若当前终端正在进行键盘输入，则不允许切换终端；若切换到终端 1，无论是从终端 1 切换到终端 1，还是从其他终端切换到终端 1，都将使 **shell** 任务开始一次命令输入，使用者可以输入 **run** 或 **end** 指令，或者直接回车取消此次指令输入
- 系统调用: `int fork()` , `int wait()` , `void exit(int)` , `int get_sem(int)` , `void free_sem(int)` , `void P(int)` , `void V(int)` .
- `printf(char* format,...)` , 和 C 的 `printf` 一样, 但只允许使用 `%s`, `%c`, `%d`, `%x` 4 种格式。

实验平台

使用 linux 系统（发行版 elementary OS）进行开发，虚拟机配置为：软盘启动，32M 内存，40M 硬盘，单核。

nasm:编译汇编语言文件，生成.o 文件

gcc:编译 c 语言文件，生成.o 文件

sublime:文本编辑器

bochs:虚拟机

系统文件概览

MyOS

| - - Makefile

| - - os.img //软盘

```
|-- boot //启动程序目录
|--|-- boot.asm //引导程序
|--|-- loader.asm //加载内核的程序
|--|-- include
|--|--|-- define.inc //常量定义
|--|--|-- fat12.inc //fat12的 BPB 头，以及常量定义
|--|--|-- loader.inc //加载地址，页表地址等常量定义
|-- include //包含目录
|--|-- all.h //包含所有头文件
|--|-- fs.h //文件系统的头文件
|--|-- hd.h //硬盘 IO 的头文件
|--|-- ipc.h //进程通信的头文件
|--|-- semaphore.h //信号量的头文件
|--|-- console.h //屏幕显示的相关结构定义和常量定义
|--|-- const.h //常量定义
|--|-- global.h //全局变量声明
|--|-- keyboard.h //键盘相关的结构定义和常量定义
|--|-- process.h //进程相关的结构定义和常量定义
|--|-- protect.h //和保护模式相关的结构定义和常量定义
|--|-- proto.h //函数声明
|--|-- sconst.inc //汇编文件使用的常量定义包含文件
|--|-- tty.h //终端相关的结构定义和常量定义
|--|-- type.h //类型定义
|-- fs //文件系统
|--|-- fs.c //文件系统任务及相关函数
|--|-- open.c //创建、打开文件相关函数
|--|-- rdwt.c //读写文件相关函数
|--|-- del.c //删除文件相关函数
|-- kernal
|--|-- hd.c //硬盘任务,硬盘相关 IO 函数
|--|-- ipc.c //进程通信任务,通信相关函数
|--|-- semaphore.c //信号量相关函数
|--|-- process.c //进程控制相关函数
|--|-- clock.c //时钟中断
|--|-- console.c //屏幕相关函数
|--|-- global.c //全局变量定义
|--|-- i8259.c //初始化8259
|--|-- kernal.asm //save, restart, 系统调用, 中断, 异常
|--|-- keyboard.c //键盘相关函数
|--|-- main.c //启动系统任务
|--|-- protect.c //初始化中断门, 异常处理
|--|-- shell.c //系统任务 shell
|--|-- start.c //初始化内核的 GDT, IDT, 中断门
|--|-- stdio.c //输入输出函数
|--|-- sys\scall.c //系统调用处理函数
|--|-- syscall.asm //系统调用
|--|-- tty.c //终端相关函数
```

```
| - - lib
| - - | - - klib.c //库
| - - | - - kliba.asm //库
| - - | - - string.asm //字符相关操作函数
| - - usr
| - - | - - user.c //用户程序
| - - | - - ulib.c //用户库
| - - | - - uliba.asm //用户库
| - - dir //linux 挂载软盘的文件夹
```

新加入功能

1. 无缓冲的进程间通信 IPC
2. 硬盘上的 inode 文件系统