系统调用的说明以及调用方式

系统调用号存放在eax寄存器中, 参数一般不超过3个，分别储存在ebx、ecx、edx寄存器中。 返回值保存在eax中。

主要参考了Linux 5.10 syscalls，详细请参见：https://man7.org/linux/man-pages/man2/syscalls.2.html ；也可以在实验用的Ubuntu虚拟机中用man命令查看，如man getdents。

系统调用execve2

```
#define __NR_execve2 87
int execve2(const char *path, char * argv[], char * envp[]);
```

- 功能：以立即加载方式执行一个指定的程序。此系统调用开始后，该进程不应再发生代码段和数据段中的缺页故障。
- 输入：
  - path: 待执行程序路径名称，
  - argv: 程序的参数，
  - envp: 环境变量的数组指针
- 返回值：成功不返回，失败返回-1；
- 其它：测试此系统调用时，请给内核打上补丁execve2.patch，以显示缺页故障和系统调用的发生。可以类似如下操作（如果想给4/linux下的源码打补丁）：

```
cd 4/linux
patch –p2 < execve2.patch
```

系统调用getdents

```
#define __NR_getdents 88
int getdents(unsigned int fd, struct linux_dirent *dirp, unsigned int
count);
```

- 功能：获取目录的目录项。
- 输入：
  - fd：所要读取目录的文件描述符。
  - dirp：一个缓存区，用于保存所读取目录的信息。缓存区的结构如下：

```
  struct linux_dirent {
    long           d_ino;
    off_t          d_off;
    unsigned short d_reclen;
```

```
    char           d_name[];
};
```

```
- count：dirp的大小。
```

- 返回值：成功执行，返回读取的字节数。当到目录结尾，则返回0。失败，则返回-1。

系统调用sleep

```
#define __NR_sleep 90
int sleep(unsigned int seconds);
```

- 功能：执行进程睡眠；
- 输入：睡眠的时间间隔；
    - seconds: 秒
- 返回值：成功返回0，失败返回-1;

系统调用getcwd

```
#define __NR_getcwd 91
long getcwd(char * buf, size_t size);
```

- 功能：获取当前工作目录；
- 输入：
    - char *buf：一块缓存区，用于保存当前工作目录的字符串。当buf设为NULL，由系统来分配缓存区。
    - size：buf缓存区的大小。
- 返回值：成功执行，则返回当前工作目录的字符串的指针。失败，则返回NULL。