

LAB5 实验报告

一 小组成员及贡献排序

汪值 林元灿 陈景昊 吕若哲 木扎拜·塔依尔

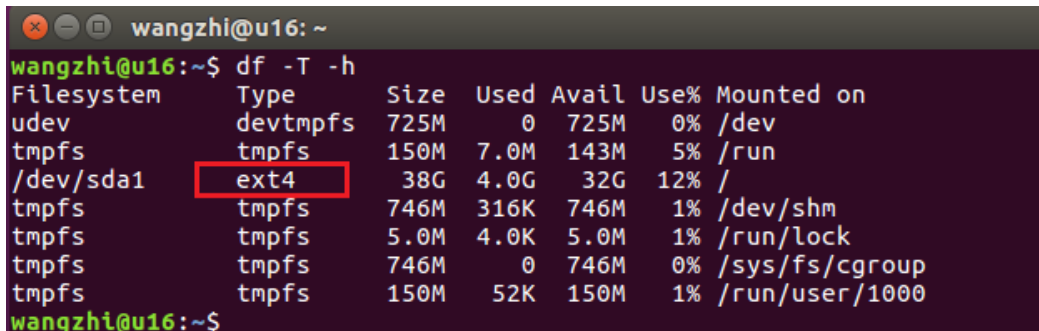
二 实验要求：

- (1) 查看当前系统所使用的文件系统的类型及版本号；
- (2) 阅读 Ext3(或 Ext4)文件系统，特别是索引节点相关的源代码。
- (3) 为内核添加一个新的系统调用 filesys, 其从调用者接收一个磁盘文件的全局路径名，打印该文件占用的所有磁盘块；
- (4) 重新编译、安装内核；
- (5) 编写用户测试程序，测试 filesys 系统调用

三 实验过程

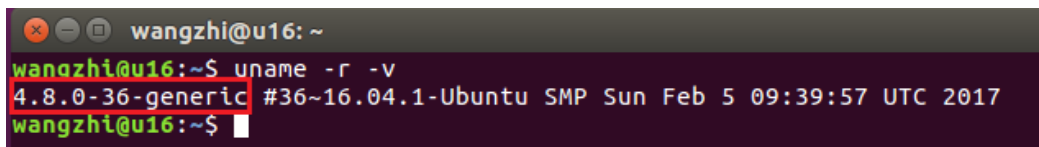
- (1) 查看当前系统所使用的文件系统的类型及版本号

文件系统类型



```
wangzhi@u16: ~  
wangzhi@u16:~$ df -T -h  
Filesystem      Type      Size  Used Avail Use% Mounted on  
udev            devtmpfs  725M    0   725M   0% /dev  
tmpfs           tmpfs     150M   7.0M   143M   5% /run  
/dev/sda1       ext4      38G   4.0G   32G  12% /  
tmpfs           tmpfs     746M  316K   746M   1% /dev/shm  
tmpfs           tmpfs     5.0M   4.0K   5.0M   1% /run/lock  
tmpfs           tmpfs     746M    0   746M   0% /sys/fs/cgroup  
tmpfs           tmpfs     150M   52K   150M   1% /run/user/1000  
wangzhi@u16:~$
```

查看版本号



```
wangzhi@u16: ~  
wangzhi@u16:~$ uname -r -v  
4.8.0-36-generic #36~16.04.1-Ubuntu SMP Sun Feb 5 09:39:57 UTC 2017  
wangzhi@u16:~$
```

- (2) 阅读 Ext3(或 Ext4)文件系统，特别是索引节点相关的源代码

Ext4 文件系统类型源码:

/fs/ext4/ext4.h

/fs/ext4/inode.c

Ext4 中的 inode 大小为 256 字节

Ext4 超级块


```

4832:     inode->i_generation = le32_to_cpu(raw_inode->i_generation);
4833:     ei->i_block_group = iloc.block_group;
4834:     ei->i_last_alloc_group = ~0;
4835:     /*
4836:      * NOTE! The in-memory inode i_data array is in little-endian order
4837:      * even on big-endian machines: we do NOT byteswap the block numbers!
4838:      */
4839:     for (block = 0; block < EXT4_N_BLOCKS; block++)
4840:         ei->i_data[block] = raw_inode->i_block[block];
4841:     INIT_LIST_HEAD(&ei->i_orphan);
4842:

```

Block 大小为 512B

(3) 为内核添加一个新的系统调用 filesys, 其从调用者接收一个磁盘文件的全局路径名, 打印该文件占用的所有磁盘块;

1) 在系统调用表中添加系统调用

a) 打开 syscalls/syscall_64.tbl 在末尾添加对应的系统调用号

327	64	preadv2	sys_preadv2
328	64	pwritev2	sys_pwritev2
329	64	fileys	sys_fileys

2) 在头文件中声明函数

a) 打开 include/linux/syscalls.h, 在末尾添加函数声明

b) asmlinkage int sys_fileys (const char __user *str);

```
asmlinkage int sys_fileys (const char __user *str);
```

3) 添加系统调用函数 位置/kernel/sys.c

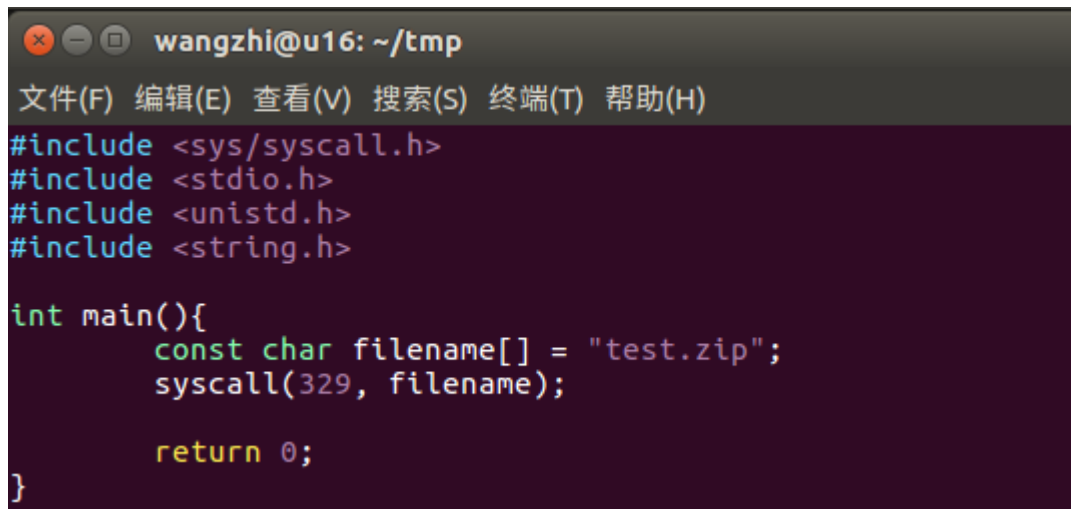
a) 实现函数 sys_fileys

```

asmlinkage int sys_fileys (const char __user *str){
    struct inode *inode=NULL;
    struct file *file=NULL;
    struct address_space i_data;
    struct radix_tree_root page_tree;
    file=filp_open(str,O_DIRECTORY,0);
    if(IS_ERR(file)){
        file=filp_open(str,O_RDONLY,0444);
        if(IS_ERR(file)){
            printk("The path %s is error!\n",str);
            return 0;
        }
        inode = file_inode(file);
        //inode=file->f_dentry->d_inode;
        printk("file name is: %s\n", str);
        printk("inode number : %ld\n",inode->i_ino);
        printk("file size(byte) : %d\n",(int)inode->i_size);
        printk("block count : %lu\n",inode->i_blocks);
        return 0;
    }
}

```

- 4) 重新编译安装内核
- 5) 添加测试文件, 运行测试文件

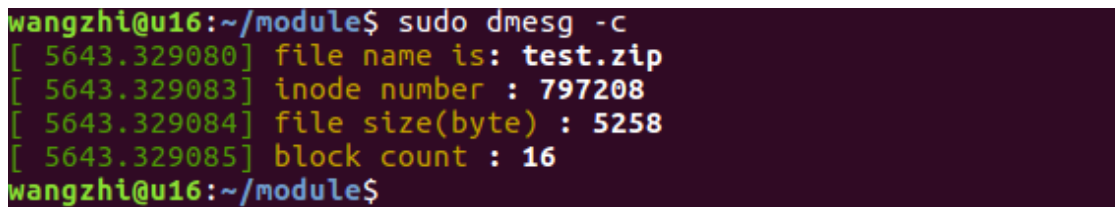


```
wangzhi@u16: ~/tmp
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
#include <sys/syscall.h>
#include <stdio.h>
#include <unistd.h>
#include <string.h>

int main(){
    const char filename[] = "test.zip";
    syscall(329, filename);

    return 0;
}
```

- 6) 查看结果



```
wangzhi@u16:~/module$ sudo dmesg -c
[ 5643.329080] file name is: test.zip
[ 5643.329083] inode number : 797208
[ 5643.329084] file size(byte) : 5258
[ 5643.329085] block count : 16
wangzhi@u16:~/module$
```