Lab 8

Chen Rui 516021910758
Wu Jiayao 517370910257

1. What is a kernel module

Modules are pieces of code that can be loaded and unloaded into the kernel upon demand.

They extend the functionality of the kernel without the need to reboot the system. But for a regular libarary to be added, you need to recompile and reboot.

2. compile a kernel

A makefile for compiling VRCow.c

```
obj-m += VRCow.o

all:
  make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules

clean:
  make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

4. mutex

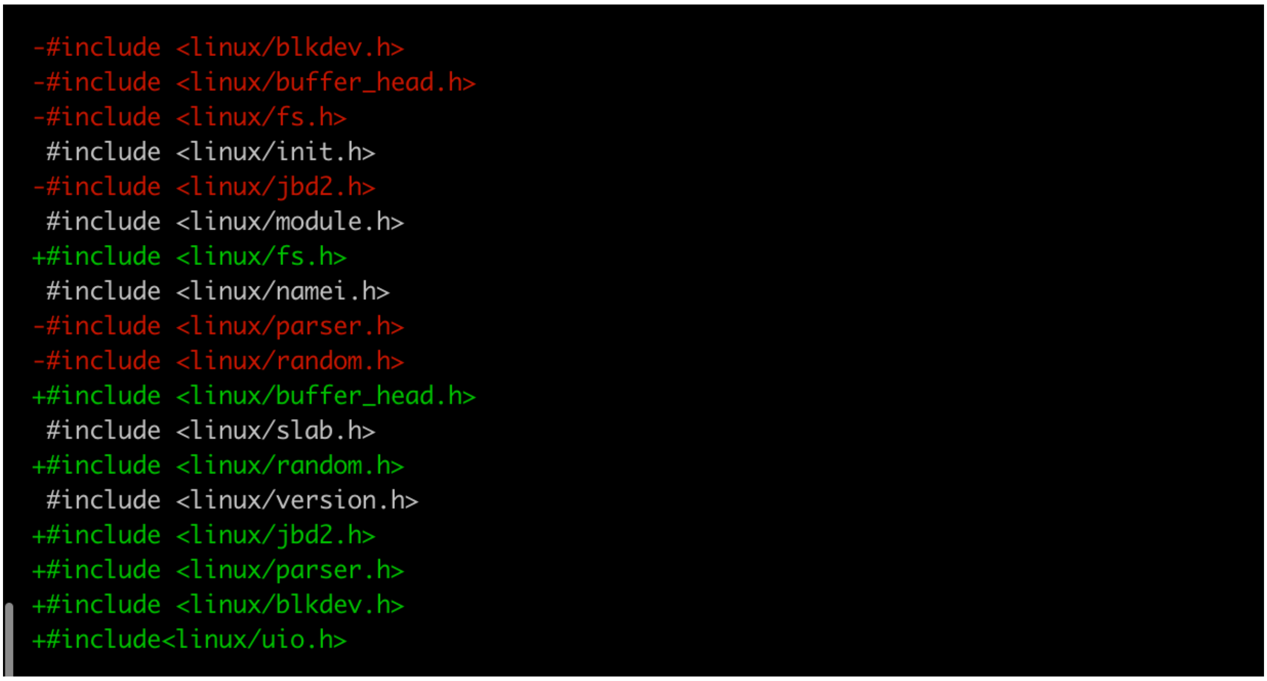
```
//define
static DEFINE_MUTEX(dadfs_sb_lock);
static DEFINE_MUTEX(dadfs_inodes_mgmt_lock);

//use
mutex_lock_interruptible(struct mutex*)

// critial region
mutex_unlock(struct mutex*)
```

I think it works properly for a filesystem, as the it can be interrupted by any signals if using **mutex_lock_interruptible**. Otherwise, operations into the filesystem may cause some trouble.

5. information

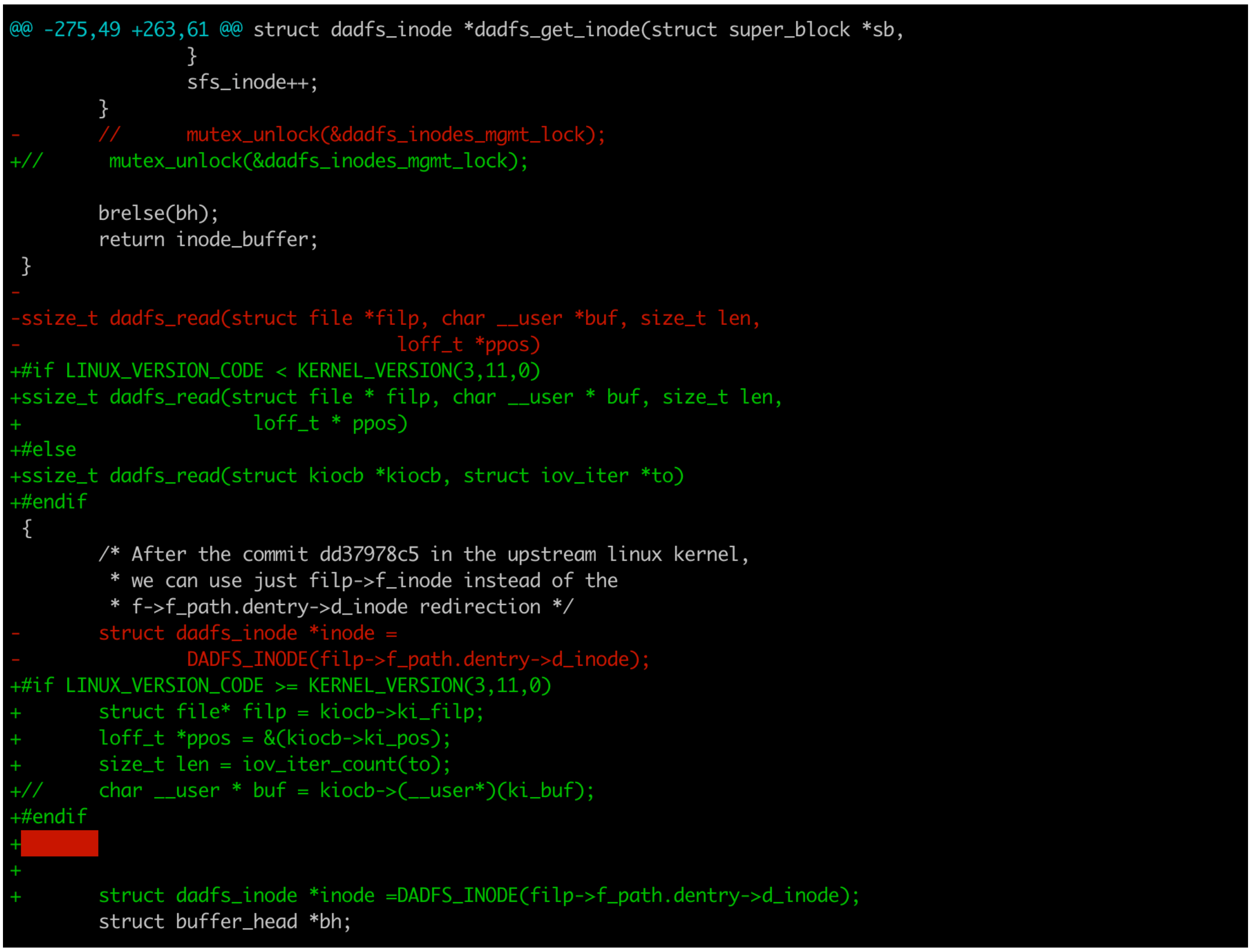
By **copy_to_iter** and **copy_from_iter**

Change

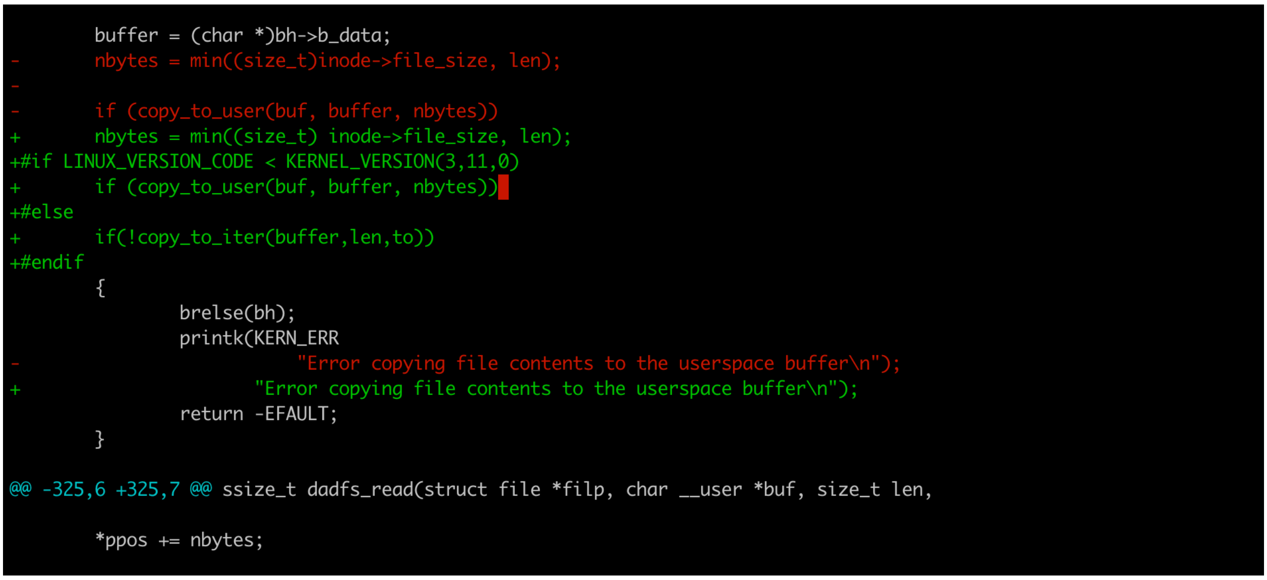


```
-#include <linux/blkdev.h>
-#include <linux/buffer_head.h>
-#include <linux/fs.h>
 #include <linux/init.h>
-#include <linux/jbd2.h>
 #include <linux/module.h>
+#include <linux/fs.h>
 #include <linux/namei.h>
-#include <linux/parser.h>
-#include <linux/random.h>
+#include <linux/buffer_head.h>
 #include <linux/slab.h>
+#include <linux/random.h>
 #include <linux/version.h>
+#include <linux/jbd2.h>
+#include <linux/parser.h>
+#include <linux/blkdev.h>
+#include<linux/uio.h>
```

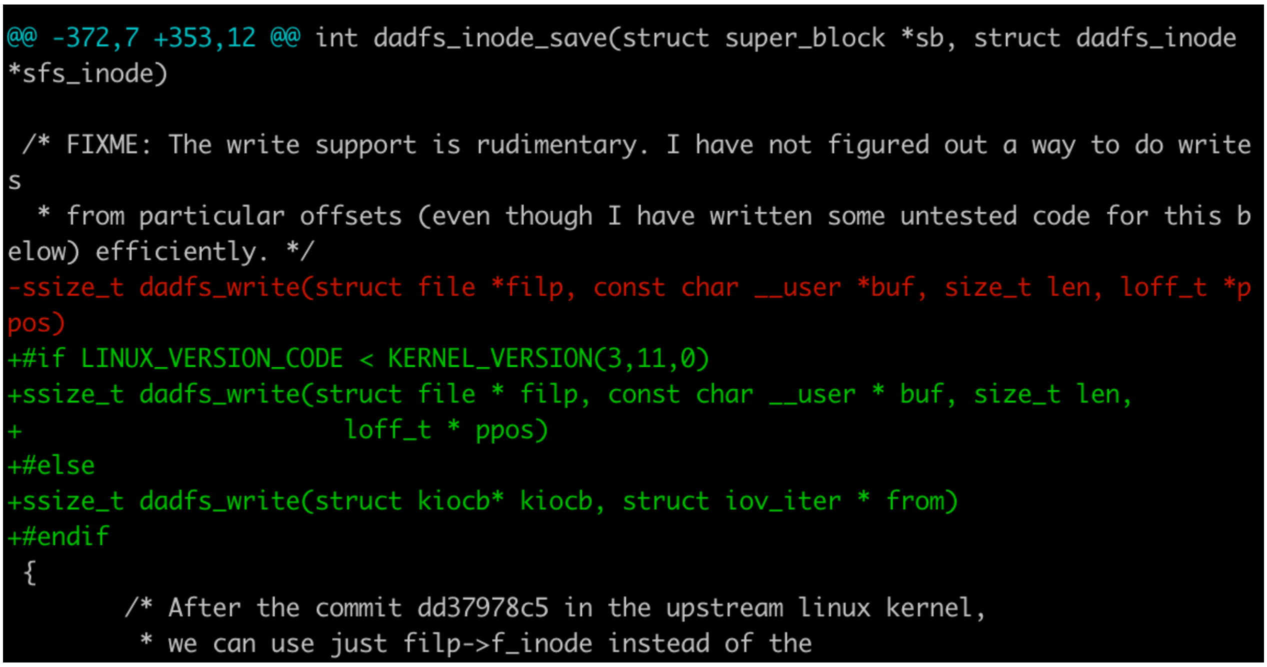
Actually only the last line is the change. Others are caused because of include file re-order by VS Code plugins



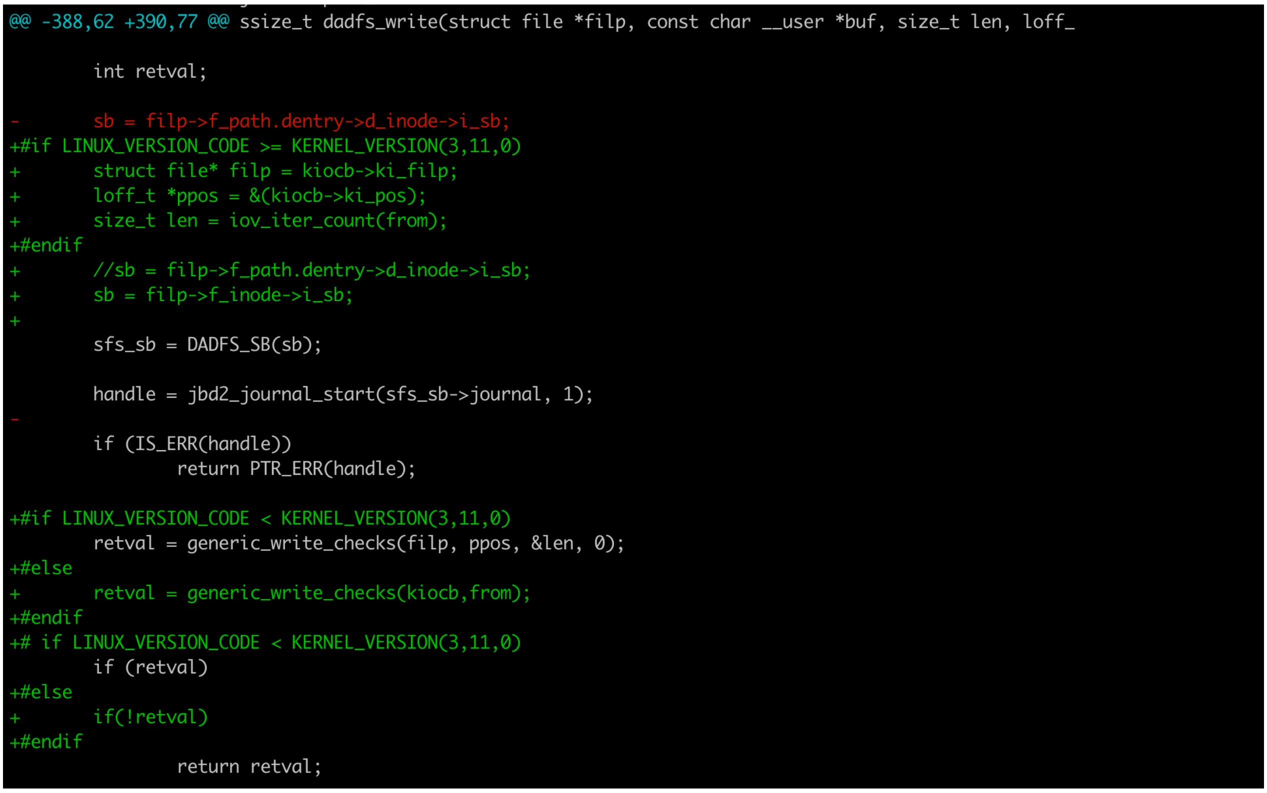
```
@@ -275,49 +263,61 @@ struct dadfs_inode *dadfs_get_inode(struct super_block *sb,
                }
                sfs_inode++;
        }
-       //      mutex_unlock(&dadfs_inodes_mgmt_lock);
+//     mutex_unlock(&dadfs_inodes_mgmt_lock);

        brelse(bh);
        return inode_buffer;
 }
-
-ssize_t dadfs_read(struct file *filp, char __user *buf, size_t len,
-                                  loff_t *ppos)
+#if LINUX_VERSION_CODE < KERNEL_VERSION(3,11,0)
+ssize_t dadfs_read(struct file * filp, char __user * buf, size_t len,
+                     loff_t * ppos)
+#else
+ssize_t dadfs_read(struct kiocb *kiocb, struct iov_iter *to)
+#endif
 {
        /* After the commit dd37978c5 in the upstream linux kernel,
         * we can use just filp->f_inode instead of the
         * f->f_path.dentry->d_inode redirection */
-       struct dadfs_inode *inode =
-               DADFS_INODE(filp->f_path.dentry->d_inode);
+#if LINUX_VERSION_CODE >= KERNEL_VERSION(3,11,0)
+       struct file* filp = kiocb->ki_filp;
+       loff_t *ppos = &(kiocb->ki_pos);
+       size_t len = iov_iter_count(to);
+//     char __user * buf = kiocb->(__user*)(ki_buf);
+#endif
+
+
+       struct dadfs_inode *inode =DADFS_INODE(filp->f_path.dentry->d_inode);
        struct buffer_head *bh;
```



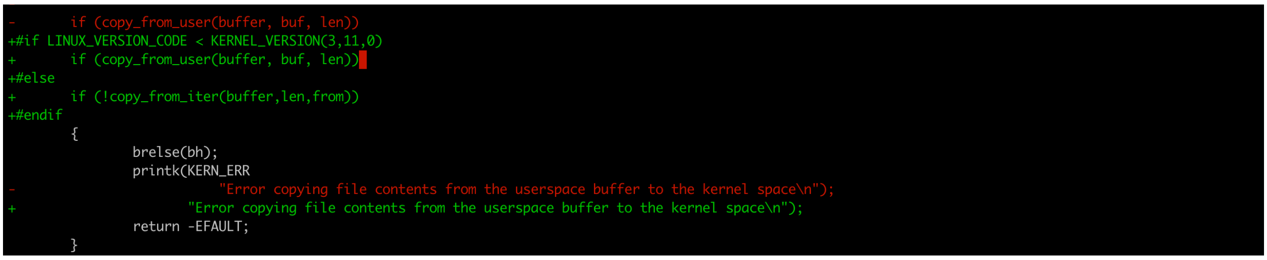
```
        buffer = (char *)bh->b_data;
-       nbytes = min((size_t)inode->file_size, len);
-
-       if (copy_to_user(buf, buffer, nbytes))
+       nbytes = min((size_t) inode->file_size, len);
+#if LINUX_VERSION_CODE < KERNEL_VERSION(3,11,0)
+       if (copy_to_user(buf, buffer, nbytes))
+#else
+       if(!copy_to_iter(buffer,len,to))
+#endif
        {
                brelse(bh);
                printk(KERN_ERR
-                               "Error copying file contents to the userspace buffer\n");
+                           "Error copying file contents to the userspace buffer\n");
                return -EFAULT;
        }

@@ -325,6 +325,7 @@ ssize_t dadfs_read(struct file *filp, char __user *buf, size_t len,

        *ppos += nbytes;
```



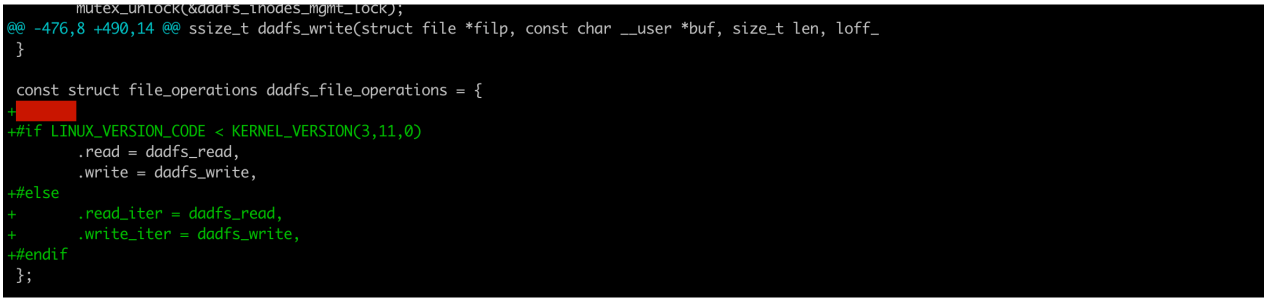
```
@@ -372,7 +353,12 @@ int dadfs_inode_save(struct super_block *sb, struct dadfs_inode
*sfs_inode)

 /* FIXME: The write support is rudimentary. I have not figured out a way to do write
s
  * from particular offsets (even though I have written some untested code for this b
elow) efficiently. */
-ssize_t dadfs_write(struct file *filp, const char __user *buf, size_t len, loff_t *p
pos)
+#if LINUX_VERSION_CODE < KERNEL_VERSION(3,11,0)
+ssize_t dadfs_write(struct file * filp, const char __user * buf, size_t len,
+                      loff_t * ppos)
+#else
+ssize_t dadfs_write(struct kiocb* kiocb, struct iov_iter * from)
+#endif
 {
        /* After the commit dd37978c5 in the upstream linux kernel,
         * we can use just filp->f_inode instead of the
```



```
@@ -388,62 +390,77 @@ ssize_t dadfs_write(struct file *filp, const char __user *buf, size_t len, loff_

        int retval;

-       sb = filp->f_path.dentry->d_inode->i_sb;
+#if LINUX_VERSION_CODE >= KERNEL_VERSION(3,11,0)
+       struct file* filp = kiocb->ki_filp;
+       loff_t *ppos = &(kiocb->ki_pos);
+       size_t len = iov_iter_count(from);
+#endif
+       //sb = filp->f_path.dentry->d_inode->i_sb;
+       sb = filp->f_inode->i_sb;
+
        sfs_sb = DADFS_SB(sb);

        handle = jbd2_journal_start(sfs_sb->journal, 1);
-
        if (IS_ERR(handle))
                return PTR_ERR(handle);

+#if LINUX_VERSION_CODE < KERNEL_VERSION(3,11,0)
        retval = generic_write_checks(filp, ppos, &len, 0);
+#else
+       retval = generic_write_checks(kiocb,from);
+#endif
+# if LINUX_VERSION_CODE < KERNEL_VERSION(3,11,0)
        if (retval)
+#else
+       if(!retval)
+#endif
                return retval;
```



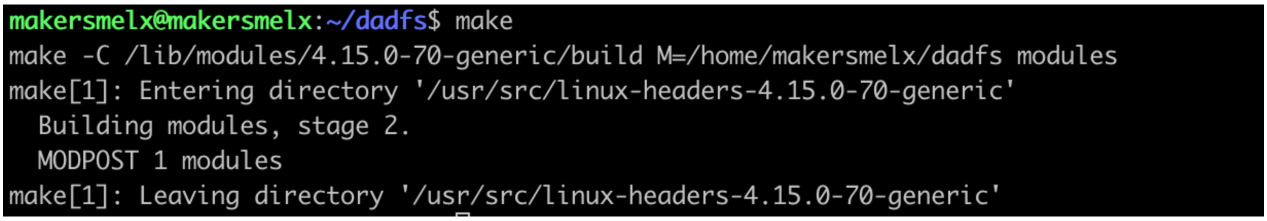
```
-       if (copy_from_user(buffer, buf, len))
+#if LINUX_VERSION_CODE < KERNEL_VERSION(3,11,0)
+       if (copy_from_user(buffer, buf, len))
+#else
+       if (!copy_from_iter(buffer,len,from))
+#endif
        {
                brelse(bh);
                printk(KERN_ERR
-                               "Error copying file contents from the userspace buffer to the kernel space\n");
+                       "Error copying file contents from the userspace buffer to the kernel space\n");
                return -EFAULT;
        }
```



```
       mutex_unlock(&dadfs_inodes_mgmt_lock);
@@ -476,8 +490,14 @@ ssize_t dadfs_write(struct file *filp, const char __user *buf, size_t len, loff_
 }

 const struct file_operations dadfs_file_operations = {
+
+#if LINUX_VERSION_CODE < KERNEL_VERSION(3,11,0)
        .read = dadfs_read,
        .write = dadfs_write,
+#else
+       .read_iter = dadfs_read,
+       .write_iter = dadfs_write,
+#endif
 };
```

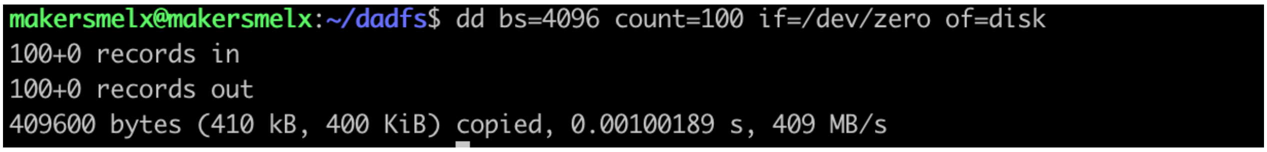
Simple process to test dadfs features:

1. compile: make



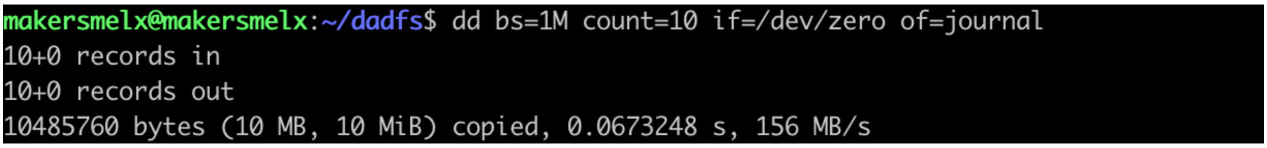
```
makersmelx@makersmelx:~/dadfs$ make
make -C /lib/modules/4.15.0-70-generic/build M=/home/makersmelx/dadfs modules
make[1]: Entering directory '/usr/src/linux-headers-4.15.0-70-generic'
  Building modules, stage 2.
  MODPOST 1 modules
make[1]: Leaving directory '/usr/src/linux-headers-4.15.0-70-generic'
```

2. create a small virtual disk (to be formated in dadfs): dd bs=4096 count=100 if=/dev/zero of=disk



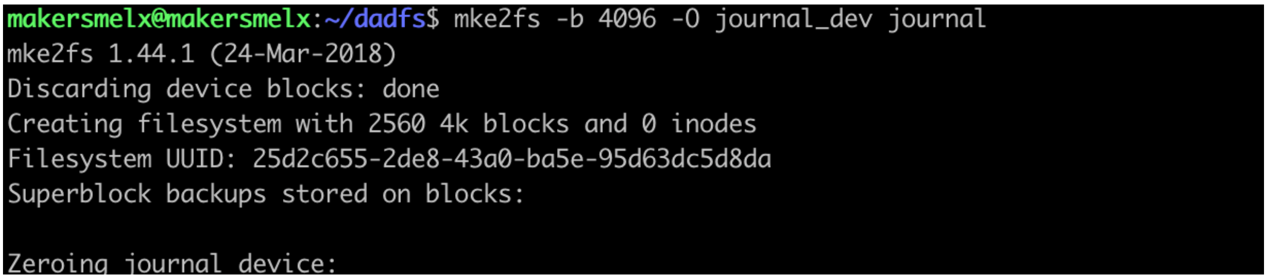
```
makersmelx@makersmelx:~/dadfs$ dd bs=4096 count=100 if=/dev/zero of=disk
100+0 records in
100+0 records out
409600 bytes (410 kB, 400 KiB) copied, 0.00100189 s, 409 MB/s
```

3. create a small virtual disk (to be used as dadfs' journal): dd bs=1M count=10 if=/dev/zero of=journal



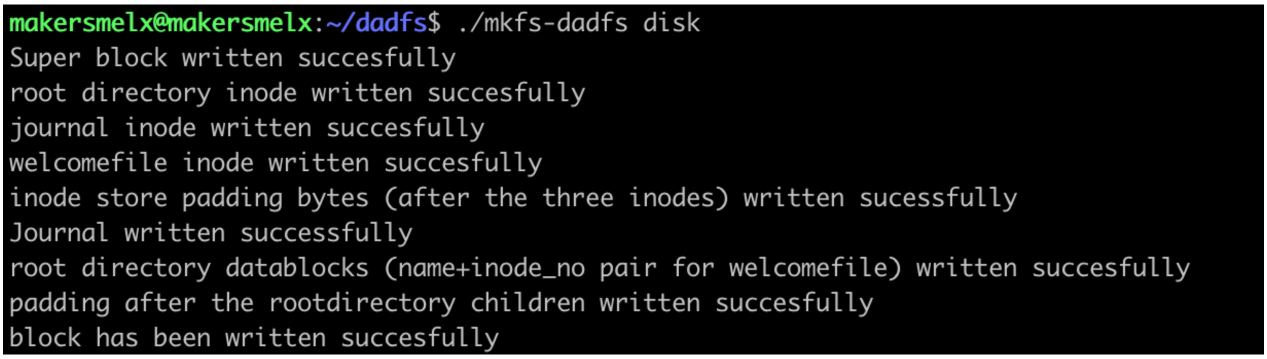
```
makersmelx@makersmelx:~/dadfs$ dd bs=1M count=10 if=/dev/zero of=journal
10+0 records in
10+0 records out
10485760 bytes (10 MB, 10 MiB) copied, 0.0673248 s, 156 MB/s
```

4. initialise the journal: mke2fs -b 4096 -O journal_dev journal



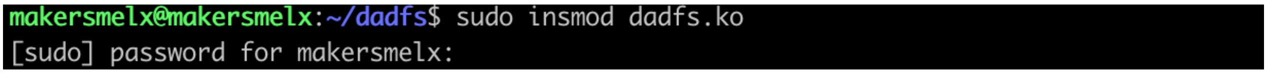
```
makersmelx@makersmelx:~/dadfs$ mke2fs -b 4096 -O journal_dev journal
mke2fs 1.44.1 (24-Mar-2018)
Discarding device blocks: done
Creating filesystem with 2560 4k blocks and 0 inodes
Filesystem UUID: 25d2c655-2de8-43a0-ba5e-95d63dc5d8da
Superblock backups stored on blocks:

Zeroing journal device:
```

5. format the disk: ./mkfs-dadfs disk



```
makersmelx@makersmelx:~/dadfs$ ./mkfs-dadfs disk
Super block written succesfully
root directory inode written succesfully
journal inode written succesfully
welcomefile inode written succesfully
inode store padding bytes (after the three inodes) written sucessfully
Journal written successfully
root directory datablocks (name+inode_no pair for welcomefile) written succesfully
padding after the rootdirectory children written succesfully
block has been written succesfully
```

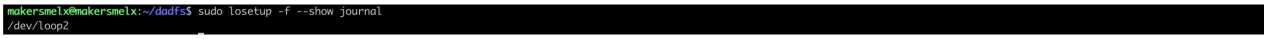
6. load dadfs module: insmod



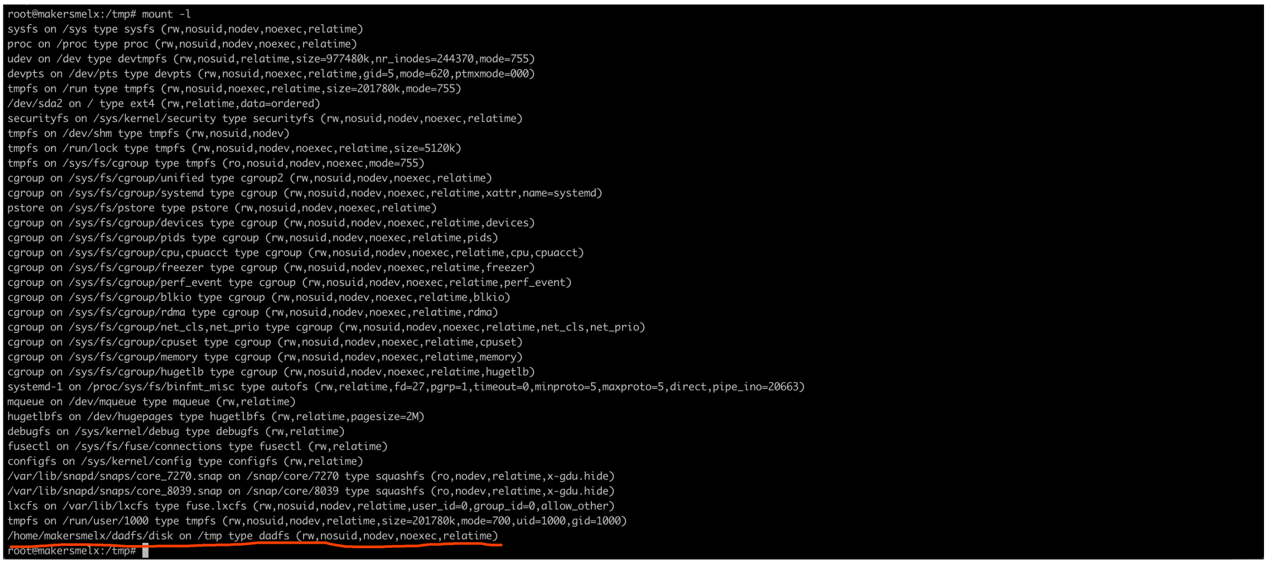
```
makersmelx@makersmelx:~/dadfs$ sudo insmod dadfs.ko
[sudo] password for makersmelx:
```

7. mount disk: losetup, mount (loop,journal_path)

```
losetup -f --show journal
sudo mount -o loop,owner,group,users,journal_path=/dev/loop2  -t dadfs disk /tmp
```



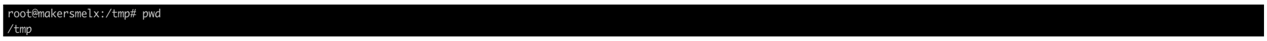
```
makersmelx@makersmelx:~/dadfs$ sudo losetup -f --show journal
/dev/loop2
```



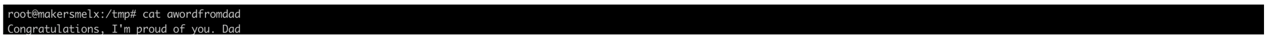
```
root@makersmelx:/tmp# mount -l
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,nosuid,relatime,size=977480k,nr_inodes=244370,mode=755)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,noexec,relatime,size=201780k,mode=755)
/dev/sda2 on / type ext4 (rw,relatime,data=ordered)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k)
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,mode=755)
cgroup on /sys/fs/cgroup/unified type cgroup2 (rw,nosuid,nodev,noexec,relatime)
cgroup on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,xattr,name=systemd)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,devices)
cgroup on /sys/fs/cgroup/pids type cgroup (rw,nosuid,nodev,noexec,relatime,pids)
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup (rw,nosuid,nodev,noexec,relatime,cpu,cpuacct)
cgroup on /sys/fs/cgroup/freezer type cgroup (rw,nosuid,nodev,noexec,relatime,freezer)
cgroup on /sys/fs/cgroup/perf_event type cgroup (rw,nosuid,nodev,noexec,relatime,perf_event)
cgroup on /sys/fs/cgroup/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,blkio)
cgroup on /sys/fs/cgroup/rdma type cgroup (rw,nosuid,nodev,noexec,relatime,rdma)
cgroup on /sys/fs/cgroup/net_cls,net_prio type cgroup (rw,nosuid,nodev,noexec,relatime,net_cls,net_prio)
cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,cpuset)
cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)
cgroup on /sys/fs/cgroup/hugetlb type cgroup (rw,nosuid,nodev,noexec,relatime,hugetlb)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=27,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=20663)
mqueue on /dev/mqueue type mqueue (rw,relatime)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,pagesize=2M)
debugfs on /sys/kernel/debug type debugfs (rw,relatime)
fusectl on /sys/fs/fuse/connections type fusectl (rw,relatime)
configfs on /sys/kernel/config type configfs (rw,relatime)
/var/lib/snapd/snaps/core_7270.snap on /snap/core/7270 type squashfs (ro,nodev,relatime,x-gdu.hide)
/var/lib/snapd/snaps/core_8039.snap on /snap/core/8039 type squashfs (ro,nodev,relatime,x-gdu.hide)
lxcfs on /var/lib/lxcfs type fuse.lxcfs (rw,nosuid,nodev,relatime,user_id=0,group_id=0,allow_other)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,size=201780k,mode=700,uid=1000,gid=1000)
/home/makersmelx/dadfs/disk on /tmp type dadfs (rw,nosuid,nodev,noexec,relatime)
root@makersmelx:/tmp# 
```

8. play with dad filesystem: mkdir, mv, cp, cat, rm, ls, cd, touch, etc.

It seems that only root user can play dadfs.

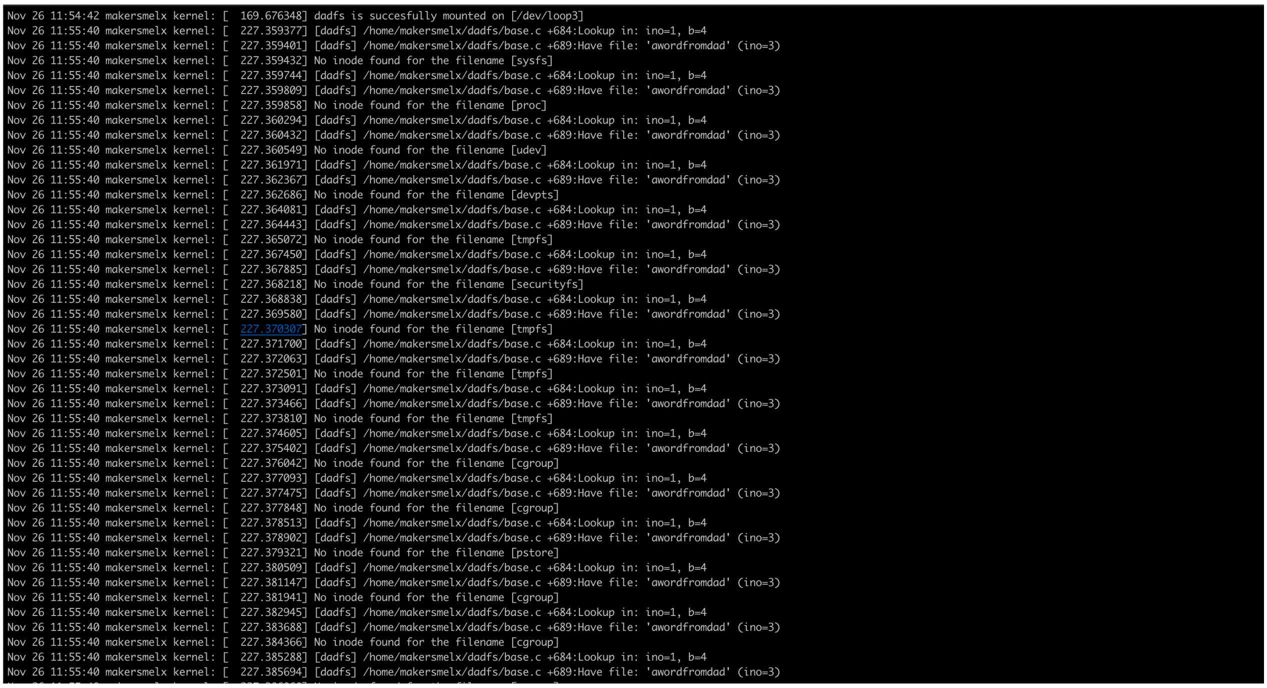


```
root@makersmelx:/tmp# pwd
/tmp
```



```
root@makersmelx:/tmp# cat awordfromdad
Congratulations, I'm proud of you. Dad
```

9. check the logs: /var/log, dmesg



```
Nov 26 11:54:42 makersmelx kernel: [  169.676348] dadfs is succesfully mounted on [/dev/loop3]
Nov 26 11:55:40 makersmelx kernel: [  227.359377] [dadfs] /home/makersmelx/dadfs/base.c +684:Lookup in: ino=1, b=4
Nov 26 11:55:40 makersmelx kernel: [  227.359401] [dadfs] /home/makersmelx/dadfs/base.c +689:Have file: 'awordfromdad' (ino=3)
Nov 26 11:55:40 makersmelx kernel: [  227.359432] No inode found for the filename [sysfs]
Nov 26 11:55:40 makersmelx kernel: [  227.359744] [dadfs] /home/makersmelx/dadfs/base.c +684:Lookup in: ino=1, b=4
Nov 26 11:55:40 makersmelx kernel: [  227.359809] [dadfs] /home/makersmelx/dadfs/base.c +689:Have file: 'awordfromdad' (ino=3)
Nov 26 11:55:40 makersmelx kernel: [  227.359858] No inode found for the filename [proc]
Nov 26 11:55:40 makersmelx kernel: [  227.360294] [dadfs] /home/makersmelx/dadfs/base.c +684:Lookup in: ino=1, b=4
Nov 26 11:55:40 makersmelx kernel: [  227.360432] [dadfs] /home/makersmelx/dadfs/base.c +689:Have file: 'awordfromdad' (ino=3)
Nov 26 11:55:40 makersmelx kernel: [  227.360549] No inode found for the filename [udev]
Nov 26 11:55:40 makersmelx kernel: [  227.361971] [dadfs] /home/makersmelx/dadfs/base.c +684:Lookup in: ino=1, b=4
Nov 26 11:55:40 makersmelx kernel: [  227.362367] [dadfs] /home/makersmelx/dadfs/base.c +689:Have file: 'awordfromdad' (ino=3)
Nov 26 11:55:40 makersmelx kernel: [  227.362686] No inode found for the filename [devpts]
Nov 26 11:55:40 makersmelx kernel: [  227.364081] [dadfs] /home/makersmelx/dadfs/base.c +684:Lookup in: ino=1, b=4
Nov 26 11:55:40 makersmelx kernel: [  227.364443] [dadfs] /home/makersmelx/dadfs/base.c +689:Have file: 'awordfromdad' (ino=3)
Nov 26 11:55:40 makersmelx kernel: [  227.365072] No inode found for the filename [tmpfs]
Nov 26 11:55:40 makersmelx kernel: [  227.367450] [dadfs] /home/makersmelx/dadfs/base.c +684:Lookup in: ino=1, b=4
Nov 26 11:55:40 makersmelx kernel: [  227.367885] [dadfs] /home/makersmelx/dadfs/base.c +689:Have file: 'awordfromdad' (ino=3)
Nov 26 11:55:40 makersmelx kernel: [  227.368218] No inode found for the filename [securityfs]
Nov 26 11:55:40 makersmelx kernel: [  227.368838] [dadfs] /home/makersmelx/dadfs/base.c +684:Lookup in: ino=1, b=4
Nov 26 11:55:40 makersmelx kernel: [  227.369580] [dadfs] /home/makersmelx/dadfs/base.c +689:Have file: 'awordfromdad' (ino=3)
Nov 26 11:55:40 makersmelx kernel: [  227.370307] No inode found for the filename [tmpfs]
Nov 26 11:55:40 makersmelx kernel: [  227.371700] [dadfs] /home/makersmelx/dadfs/base.c +684:Lookup in: ino=1, b=4
Nov 26 11:55:40 makersmelx kernel: [  227.372063] [dadfs] /home/makersmelx/dadfs/base.c +689:Have file: 'awordfromdad' (ino=3)
Nov 26 11:55:40 makersmelx kernel: [  227.372501] No inode found for the filename [tmpfs]
Nov 26 11:55:40 makersmelx kernel: [  227.373091] [dadfs] /home/makersmelx/dadfs/base.c +684:Lookup in: ino=1, b=4
Nov 26 11:55:40 makersmelx kernel: [  227.373466] [dadfs] /home/makersmelx/dadfs/base.c +689:Have file: 'awordfromdad' (ino=3)
Nov 26 11:55:40 makersmelx kernel: [  227.373810] No inode found for the filename [tmpfs]
Nov 26 11:55:40 makersmelx kernel: [  227.374605] [dadfs] /home/makersmelx/dadfs/base.c +684:Lookup in: ino=1, b=4
Nov 26 11:55:40 makersmelx kernel: [  227.375402] [dadfs] /home/makersmelx/dadfs/base.c +689:Have file: 'awordfromdad' (ino=3)
Nov 26 11:55:40 makersmelx kernel: [  227.376042] No inode found for the filename [cgroup]
Nov 26 11:55:40 makersmelx kernel: [  227.377093] [dadfs] /home/makersmelx/dadfs/base.c +684:Lookup in: ino=1, b=4
Nov 26 11:55:40 makersmelx kernel: [  227.377475] [dadfs] /home/makersmelx/dadfs/base.c +689:Have file: 'awordfromdad' (ino=3)
Nov 26 11:55:40 makersmelx kernel: [  227.377848] No inode found for the filename [cgroup]
Nov 26 11:55:40 makersmelx kernel: [  227.378513] [dadfs] /home/makersmelx/dadfs/base.c +684:Lookup in: ino=1, b=4
Nov 26 11:55:40 makersmelx kernel: [  227.378902] [dadfs] /home/makersmelx/dadfs/base.c +689:Have file: 'awordfromdad' (ino=3)
Nov 26 11:55:40 makersmelx kernel: [  227.379321] No inode found for the filename [pstore]
Nov 26 11:55:40 makersmelx kernel: [  227.380509] [dadfs] /home/makersmelx/dadfs/base.c +684:Lookup in: ino=1, b=4
Nov 26 11:55:40 makersmelx kernel: [  227.381147] [dadfs] /home/makersmelx/dadfs/base.c +689:Have file: 'awordfromdad' (ino=3)
Nov 26 11:55:40 makersmelx kernel: [  227.381941] No inode found for the filename [cgroup]
Nov 26 11:55:40 makersmelx kernel: [  227.382945] [dadfs] /home/makersmelx/dadfs/base.c +684:Lookup in: ino=1, b=4
Nov 26 11:55:40 makersmelx kernel: [  227.383688] [dadfs] /home/makersmelx/dadfs/base.c +689:Have file: 'awordfromdad' (ino=3)
Nov 26 11:55:40 makersmelx kernel: [  227.384366] No inode found for the filename [cgroup]
Nov 26 11:55:40 makersmelx kernel: [  227.385288] [dadfs] /home/makersmelx/dadfs/base.c +684:Lookup in: ino=1, b=4
Nov 26 11:55:40 makersmelx kernel: [  227.385694] [dadfs] /home/makersmelx/dadfs/base.c +689:Have file: 'awordfromdad' (ino=3)
```

10. umount disk: losetup, umount

11. unload module: rmmod