

README

- We are attaching 4 files - 'mykmod_main.c', 'memutil.cpp', 'REPORT.pdf', 'README.pdf' in a folder.

For building and loading the driver:

- first go to the directory and do make.
- Now to load the kernel modules into the kernel, use the command 'insmod'.

Ex: `insmod kernel/mykmod.ko`

- To create a device special file use the command 'mknod'.

Ex: `mknod /tmp/mydev_pR6 c 243 10`

- Figures are attached below for more details.

Building and loading the driver:

```
[root@cs3523 ~]# cd 99_devmmap_paging
[root@cs3523 99_devmmap_paging]# make
make[1]: Entering directory `/root/99_devmmap_paging/kernel'
make -C /lib/modules/3.10.0-1062.9.1.el7.x86_64/build M=/root/99_devmmap_paging/kernel modules
make[2]: Entering directory `/usr/src/kernels/3.10.0-1062.9.1.el7.x86_64'
CC [M] /root/99_devmmap_paging/kernel/mykmod_main.o
LD [M] /root/99_devmmap_paging/kernel/mykmod.o
Building modules, stage 2.
MODPOST 1 modules
CC /root/99_devmmap_paging/kernel/mykmod.mod.o
LD [M] /root/99_devmmap_paging/kernel/mykmod.ko
make[2]: Leaving directory `/usr/src/kernels/3.10.0-1062.9.1.el7.x86_64'
make[1]: Leaving directory `/root/99_devmmap_paging/kernel'
make[1]: Entering directory `/root/99_devmmap_paging/util'
g++ -std=c++11 -I ../include -I ../lib -L ../lib -lrt -o memutil memutil.cpp -g -lpthread
make[1]: Leaving directory `/root/99_devmmap_paging/util'
[root@cs3523 99_devmmap_paging]# insmod kernel/mykmod.ko
[root@cs3523 99_devmmap_paging]# grep mykmod /proc/devices
243 mykmod
```

Unloading the driver:

- Use the commands, "rm -f /tmp/mydev*" - to remove the files.
- And "rmmod mykmod" - to remove the module.

To run: use the following commands

- “./util/memutil /tmp/mydev_pR6 --pt prefetch --op mapread” - In case of prefetch read.
- “./util/memutil /tmp/mydev_JZ1 --pt demand --op mapread” - In case of demand paging read.
- “./util/memutil /tmp/mydev_fBc --pt prefetch --op mapwrite --op mapread --mes test2” - In case of prefetch write and read.
- “./util/memutil /tmp/mydev_Ln5 --pt demand --op mapwrite --op mapread --mes test2” - In case of demand paging write and read.

SAMPLE OUTPUTS:

1. Sample output in case of a prefetch read is shown in the figure below.

```
[root@cs3523 99_devmmap_paging]# mknod /tmp/mydev_pR6 c 243 10
[root@cs3523 99_devmmap_paging]# ./util/memutil /tmp/mydev_pR6 --pt prefetch --op mapread
[root@cs3523 99_devmmap_paging]# dmesg | grep -e mykmod_vm_open -e mykmod_vm_close
[17700.695910] mykmod_vm_open: vma=ffff9efdf4e58438 npagefaults:0
[17700.711044] mykmod_vm_close: vma=ffff9efdf4e58438 npagefaults:256
```

Total no.of page faults occurred = 256.

2. Sample output in case of a demand paging read is shown in the figure below.

```
[root@cs3523 99_devmmap_paging]# mknod /tmp/mydev_JZ1 c 243 11
[root@cs3523 99_devmmap_paging]# ./util/memutil /tmp/mydev_JZ1 --pt demand --op mapread
[root@cs3523 99_devmmap_paging]# dmesg | grep -e mykmod_vm_open -e mykmod_vm_close
[17700.695910] mykmod_vm_open: vma=ffff9efdf4e58438 npagefaults:0
[17700.711044] mykmod_vm_close: vma=ffff9efdf4e58438 npagefaults:256
[17808.506074] mykmod_vm_open: vma=ffff9efe77285bd0 npagefaults:0
[17808.521132] mykmod_vm_close: vma=ffff9efe77285bd0 npagefaults:256
```

Here the last two lines of dmesg are corresponding to the output.
So the total no.of page faults occurred = 256.

3. Sample output in case of a prefetch write and read is shown in the figure below.

```
[root@cs3523 99_devmmap_paging]# mknod /tmp/mydev_f8c c 243 20
[root@cs3523 99_devmmap_paging]# ./util/memutil /tmp/mydev_f8c --pt prefetch --op mapwrite --op mapread --mes test2
[root@cs3523 99_devmmap_paging]# dmesg | grep -e mykmod_vm_open -e mykmod_vm_close
[17700.695910] mykmod_vm_open: vma=ffff9efdf4e58438 npagefaults:0
[17700.711044] mykmod_vm_close: vma=ffff9efdf4e58438 npagefaults:256
[17808.506074] mykmod_vm_open: vma=ffff9efe77285bd0 npagefaults:0
[17808.521132] mykmod_vm_close: vma=ffff9efe77285bd0 npagefaults:256
[17892.502833] mykmod_vm_open: vma=ffff9efdfa274a20 npagefaults:0
[17892.531302] mykmod_vm_close: vma=ffff9efdfa274a20 npagefaults:256
[17892.531349] mykmod_vm_open: vma=ffff9efdfa274a20 npagefaults:0
[17892.540685] mykmod_vm_close: vma=ffff9efdfa274a20 npagefaults:256
```

Here the last four lines of dmesg are corresponding to the output.

So the total no. of page faults occurred = 256,256 in both read and write.

4. Sample output in case of a demand paging write and read is shown in the figure below.

```
[root@cs3523 99_devmmap_paging]# mknod /tmp/mydev_Ln5 c 243 21
[root@cs3523 99_devmmap_paging]# ./util/memutil /tmp/mydev_Ln5 --pt demand --op mapwrite --op mapread --mes test2
[root@cs3523 99_devmmap_paging]# dmesg | grep -e mykmod_vm_open -e mykmod_vm_close
[17700.695910] mykmod_vm_open: vma=ffff9efdf4e58438 npagefaults:0
[17700.711044] mykmod_vm_close: vma=ffff9efdf4e58438 npagefaults:256
[17808.506074] mykmod_vm_open: vma=ffff9efe77285bd0 npagefaults:0
[17808.521132] mykmod_vm_close: vma=ffff9efe77285bd0 npagefaults:256
[17892.502833] mykmod_vm_open: vma=ffff9efdfa274a20 npagefaults:0
[17892.531302] mykmod_vm_close: vma=ffff9efdfa274a20 npagefaults:256
[17892.531349] mykmod_vm_open: vma=ffff9efdfa274a20 npagefaults:0
[17892.540685] mykmod_vm_close: vma=ffff9efdfa274a20 npagefaults:256
[17988.370905] mykmod_vm_open: vma=ffff9efdf4e58d80 npagefaults:0
[17988.394740] mykmod_vm_close: vma=ffff9efdf4e58d80 npagefaults:256
[17988.394752] mykmod_vm_open: vma=ffff9efdf4e58d80 npagefaults:0
[17988.404241] mykmod_vm_close: vma=ffff9efdf4e58d80 npagefaults:256
```

Here the last four lines of dmesg are corresponding to the output.

So the total no. of page faults occurred = 256,256 in both read and write.

Observation: The no. of page faults in both cases are the same. Only difference is in case of prefetching, all the page faults for the entire 1MB are generated in the context of mmap itself, whereas in case of demand paging page faults are generated when the application starts reading/writing from/to the memory.