**实 验 报 告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **组 别** |  | **姓 名** | 尹浩男 | **同组实验者** |  |
| **实验项目名称** | Linux驱动程序 | | | **实验日期** | 12月 28 日 |
| **教师评语:** |  | | | | |
| **实验成绩：** | | | **指导教师（签名）：**  2020年 月 日 | | |
| **一．实验目的**  1. 掌握Linux设备驱动程序的基本原理  2. 掌握Linux字符设备驱动的创建流程  3. 掌握Linux字符设备驱动的常见函数  4. 理解Linux设备驱动的内存映射机制  **二．实验内容**  mmapdriver内核模块测试时，只显示了内存映射区域的信息。如果想把当前进程在内存映射后vma区域的变化显示出来，应该如何测试？  **三. 实验步骤**   |  | | --- | | 第一步 make  root@yhnComputer:/yhnCode/driver# make  make -C /lib/modules/5.4.0-58-generic/build M=/yhnCode/driver modules  make[1]: Entering directory '/usr/src/linux-headers-5.4.0-58-generic'    CC [M]  /yhnCode/driver/miscdev\_map.o    Building modules, stage 2.    MODPOST 1 modules    CC [M]  /yhnCode/driver/miscdev\_map.mod.o    LD [M]  /yhnCode/driver/miscdev\_map.ko  make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-58-generic'  rm -r -f .tmp\_versions \*.mod.c .\*.cmd \*.o \*.symvers |   **四. 实验结果**   |  | | --- | | root@yhnComputer:/yhnCode/vma# ./test  This is parent pid 33048  Info from the Memory Management structure for task 'test' (pid=33048)     pgd=FFFF8FA6462A8000  mmap=FFFF8FA5956C8D00  map\_count=22  mm\_users=1  mm\_count=1      start\_code=55B795740000   end\_code=55B795741545      start\_data=55B795743D60   end\_data=55B795744010       start\_brk=55B795A87000        brk=55B795AA8000       arg\_start=7FFDF7B471A1    arg\_end=7FFDF7B471A8       env\_start=7FFDF7B471A8    env\_end=7FFDF7B47FF1     start\_stack=7FFDF7B45B50    down\_to=7FFDF7B24B50  <--- stack grows downward  List of the Virtual Memory Areas for task 'test' (pid=33048)    1  vm\_start=55B795740000  vm\_end=55B795741000  r--p    2  vm\_start=55B795741000  vm\_end=55B795742000  r-xp    3  vm\_start=55B795742000  vm\_end=55B795743000  r--p    4  vm\_start=55B795743000  vm\_end=55B795744000  r--p    5  vm\_start=55B795744000  vm\_end=55B795745000  rw-p    6  vm\_start=55B795A87000  vm\_end=55B795AA8000  rw-p    7  vm\_start=7FCB5F3C7000  vm\_end=7FCB5F3EC000  r--p    8  vm\_start=7FCB5F3EC000  vm\_end=7FCB5F564000  r-xp    9  vm\_start=7FCB5F564000  vm\_end=7FCB5F5AE000  r--p   10  vm\_start=7FCB5F5AE000  vm\_end=7FCB5F5AF000  ---p   11  vm\_start=7FCB5F5AF000  vm\_end=7FCB5F5B2000  r--p   12  vm\_start=7FCB5F5B2000  vm\_end=7FCB5F5B5000  rw-p   13  vm\_start=7FCB5F5B5000  vm\_end=7FCB5F5BB000  rw-p   14  vm\_start=7FCB5F5CD000  vm\_end=7FCB5F5CE000  r--p   15  vm\_start=7FCB5F5CE000  vm\_end=7FCB5F5F1000  r-xp   16  vm\_start=7FCB5F5F1000  vm\_end=7FCB5F5F9000  r--p   17  vm\_start=7FCB5F5FA000  vm\_end=7FCB5F5FB000  r--p   18  vm\_start=7FCB5F5FB000  vm\_end=7FCB5F5FC000  rw-p   19  vm\_start=7FCB5F5FC000  vm\_end=7FCB5F5FD000  rw-p   20  vm\_start=7FFDF7B27000  vm\_end=7FFDF7B48000  rw-p   21  vm\_start=7FFDF7B51000  vm\_end=7FFDF7B54000  r--p   22  vm\_start=7FFDF7B54000  vm\_end=7FFDF7B55000  r-xp  CR3=1062A8006  mm->pgd=000000002c2127af  mm->map\_count=22  This is child pid 33049  Info from the Memory Management structure for task 'test' (pid=33049)     pgd=FFFF8FA648E28000  mmap=FFFF8FA5956C96C0  map\_count=22  mm\_users=1  mm\_count=1      start\_code=55B795740000   end\_code=55B795741545      start\_data=55B795743D60   end\_data=55B795744010       start\_brk=55B795A87000        brk=55B795AA8000       arg\_start=7FFDF7B471A1    arg\_end=7FFDF7B471A8       env\_start=7FFDF7B471A8    env\_end=7FFDF7B47FF1     start\_stack=7FFDF7B45B50    down\_to=7FFDF7B24B50  <--- stack grows downward  List of the Virtual Memory Areas for task 'test' (pid=33049)    1  vm\_start=55B795740000  vm\_end=55B795741000  r--p    2  vm\_start=55B795741000  vm\_end=55B795742000  r-xp    3  vm\_start=55B795742000  vm\_end=55B795743000  r--p    4  vm\_start=55B795743000  vm\_end=55B795744000  r--p    5  vm\_start=55B795744000  vm\_end=55B795745000  rw-p    6  vm\_start=55B795A87000  vm\_end=55B795AA8000  rw-p    7  vm\_start=7FCB5F3C7000  vm\_end=7FCB5F3EC000  r--p    8  vm\_start=7FCB5F3EC000  vm\_end=7FCB5F564000  r-xp    9  vm\_start=7FCB5F564000  vm\_end=7FCB5F5AE000  r--p   10  vm\_start=7FCB5F5AE000  vm\_end=7FCB5F5AF000  ---p   11  vm\_start=7FCB5F5AF000  vm\_end=7FCB5F5B2000  r--p   12  vm\_start=7FCB5F5B2000  vm\_end=7FCB5F5B5000  rw-p   13  vm\_start=7FCB5F5B5000  vm\_end=7FCB5F5BB000  rw-p   14  vm\_start=7FCB5F5CD000  vm\_end=7FCB5F5CE000  r--p   15  vm\_start=7FCB5F5CE000  vm\_end=7FCB5F5F1000  r-xp   16  vm\_start=7FCB5F5F1000  vm\_end=7FCB5F5F9000  r--p   17  vm\_start=7FCB5F5FA000  vm\_end=7FCB5F5FB000  r--p   18  vm\_start=7FCB5F5FB000  vm\_end=7FCB5F5FC000  rw-p   19  vm\_start=7FCB5F5FC000  vm\_end=7FCB5F5FD000  rw-p   20  vm\_start=7FFDF7B27000  vm\_end=7FFDF7B48000  rw-p   21  vm\_start=7FFDF7B51000  vm\_end=7FFDF7B54000  r--p   22  vm\_start=7FFDF7B54000  vm\_end=7FFDF7B55000  r-xp  CR3=108E28002  mm->pgd=00000000cdae4459  mm->map\_count=22 |   实验结果如下图所示：  **五. 分析**  Linux设备管理的核心思想是以文件方式实现对设备的管理，即设备文件，设备文件在建立过程中，需要设备号，需要与设备文件相关联的文件操作函数集合，从本质上看，这些与设备文件相关联的文件操作函数实际构成了设备驱动程序的实现。  在本次实验中，以读写的方式打开设备/dev/mymap，然后调用mmap函数对打开的文件描述符fd进行内存映射，因为要测试该进程在内存映射前后vma区域的变化，所以，我们需要在进行内存映射前和进行内存映射后，分别读取vma区域的信息，我们可以通过vma内核模块，来读取到vma区域的信息。 | | | | | |