

操作系统课程设计实验报告

实验题目：实验二 Linux 内核模块编程之进程树 pstree

姓 名：张孜远

学 号：20151521

组 号：04

专 业：卓越学院 智能计算与数据科学

班 级：20186211

老师姓名：任彧老师

日 期：2022 年 12 月 20 日

目 录

一 题目介绍.....	1
二 实验内容与思路.....	1
三 遇到问题及解决方法.....	1
四 核心代码及实验结果展示	1
五 个人实验改进与总结.....	4
5.1 个人实验小结	4
5.2 个人实验总结	4
六 参考文献.....	4

一 题目介绍

实验目的：

Linux 提供的模块机制能动态扩充 Linux 功能，无需重新编译内核，已经广泛应用于 Linux 内核的许多功能的实现中。在本实验中将学习模块的基本概念、原理及实现技术，然后利用内核模块编程访问进程的基本信息，加深对进程概念的理解，掌握基本的模块编程技术。

二 实验内容与思路

实验内容：

(1)设计一个模块，要求列出系统中所有内核线程的程序名、PID、进程状态、进程优先级、父进程的 PID；

(2)设计一个带参数的模块，其参数为某个进程的 PID 号，模块的功能是列出该进程的家庭信息，包括父进程、兄弟进程和子进程的程序名、PID 号及进程状态；

(3)阅读并分析 Linux 内核函数的源码。

考察的知识点：Linux 内核模块编程、Linux 进程管理方法等。

三 遇到问题及解决方法

遇到的问题：

1、printf 函数无法正常输出信息：

原因：由于系统默认是行缓存，所以是没有换行导致的错误。

实验方法：

在虚拟机中 Linux 环境下使用 C 语言进行编程，参照书本与课堂要求进行 Linux 模块编程并安装，最后用 dmesg 指令验证。

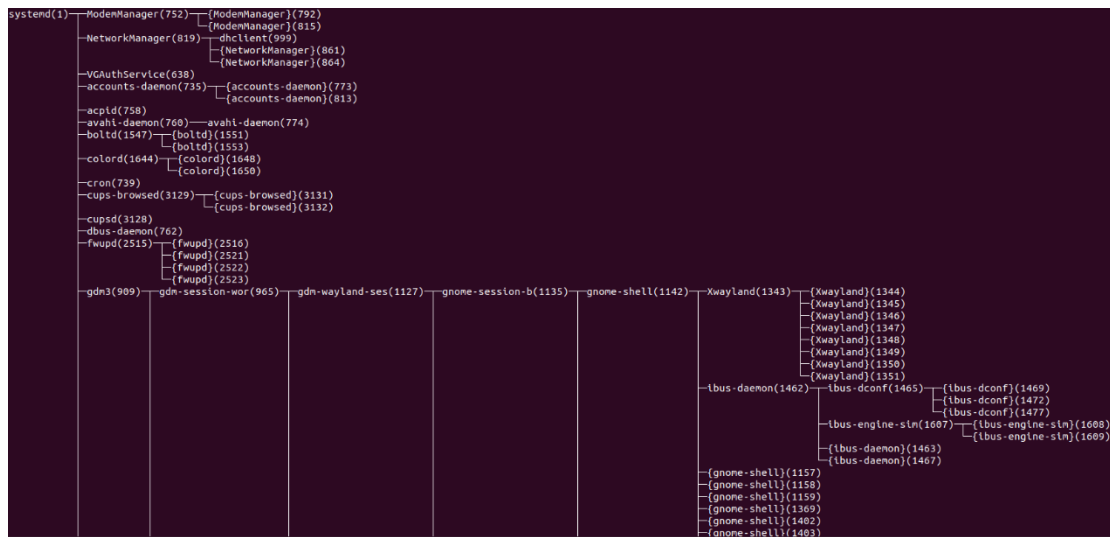
四 核心代码及实验结果展示

实验过程和结果：

第一步：进入 root 用户模式下，执行 make 指令：

```
zhangziyuan@ubuntu:~/exp2$ su root
Password:
su: Authentication failure
zhangziyuan@ubuntu:~/exp2$ su root
Password:
root@ubuntu:/home/zhangziyuan/exp2# make
make -C /lib/modules/5.4.0-135-generic/build M=/home/zhangziyuan/exp2 modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-135-generic'
Building modules, stage 2.
MODPOST 1 modules
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-135-generic'
root@ubuntu:/home/zhangziyuan/exp2#
```

第二步：通过 `pstree -p` 指令选择合适的进程（合适的定义：该进程有父进程、兄弟进程和子进程）



上图所示的是 `pstree` 树的部分全貌：

由上图截取部分进程树如下，可知：pid=760 号的进程是符合条件的合适进程。

```
root@ubuntu:/home/zhangziyuan/exp2# pstree -p
systemd(1)─ModemManager(752)─{ModemManager}(792)
               │               └─{ModemManager}(815)
               └─NetworkManager(819)─dhclient(999)
                                   └─{NetworkManager}(861)
                                       └─{NetworkManager}(864)
               └─VGAuthService(638)
               └─accounts-daemon(735)─{accounts-daemon}(773)
                                       └─{accounts-daemon}(813)
               └─acpid(758)
               └─avahi-daemon(760)─avahi-daemon(774)
               └─boltd(1547)─{boltd}(1551)
                           └─{boltd}(1553)
               └─colord(1644)─{colord}(1648)
                           └─{colord}(1650)
               └─cron(739)
               └─cups-browsed(3129)─{cups-browsed}(3131)
                                   └─{cups-browsed}(3132)
               └─cupsd(3128)
               └─dbus-daemon(762)
               └─fwupd(2515)─{fwupd}(2516)
                           └─{fwupd}(2521)
                               └─{fwupd}(2522)
                                   └─{fwupd}(2523)
               └─gdn3(909)─gdn-session-wor(965)─gdn-wayland-ses(1127)─gnome-session-b(1135)─gnome-shell(1142)─Xwayland(1343)─{Xwayland}(1344)
                                                           └─{Xwayland}(1345)
                                                           └─{Xwayland}(1346)
                                                           └─{Xwayland}(1347)
                                                           └─{Xwayland}(1348)
                                                           └─{Xwayland}(1349)
                                                           └─{Xwayland}(1350)
                                                           └─{Xwayland}(1351)
                                                           └─ibus-daemon(1462)─ibus-dconf(1465)─{ibus-dconf}(1469)
                                                           │               └─{ibus-dconf}(1472)
                                                           │               └─{ibus-dconf}(1477)
                                                           └─ibus-engine-sln(1607)─{ibus-engine-sln}(1608)
                                                           │               └─{ibus-engine-sln}(1609)
                                                           └─{gnome-shell}(1157)
                                                           └─{gnome-shell}(1158)
                                                           └─{gnome-shell}(1159)
                                                           └─{gnome-shell}(1369)
                                                           └─{gnome-shell}(1402)
                                                           └─{gnome-shell}(1403)
```

第三步：通过 `insmod` 指令安装模块，使用 `dmesg` 指令查看输出的结果：

```
root@ubuntu:/home/zhangziyuan/exp2# insmod mypstree.ko pid=760
root@ubuntu:/home/zhangziyuan/exp2# dmesg
```

第四步：dmesg 的结果如下：

```
[ 8233.260386] systemd(pid:1,state:1)---avahi-daemon(pid:760,state:1)---avahi-daemon(pid:774,state:1)
[ 8233.260387] |systemd-journal(pid:423,state:1)
[ 8233.260387] |systemd-udev(pid:473,state:1)
[ 8233.260388] |vmware-vmblock-(pid:496,state:1)
[ 8233.260388] |systemd-resolve(pid:590,state:1)
[ 8233.260389] |systemd-timesyn(pid:593,state:1)
[ 8233.260390] |VGAuthService(pid:638,state:1)
[ 8233.260390] |vmttoolsd(pid:646,state:1)
[ 8233.260391] |accounts-daemon(pid:735,state:1)
[ 8233.260391] |cron(pid:739,state:1)
[ 8233.260392] |networkd-dispat(pid:745,state:1)
[ 8233.260392] |ModemManager(pid:752,state:1)
[ 8233.260393] |acpid(pid:758,state:1)
[ 8233.260393] |udisksd(pid:759,state:1)
[ 8233.260394] |dbus-daemon(pid:762,state:1)
[ 8233.260394] |wpa_supplicant(pid:812,state:1)
[ 8233.260394] |rsyslogd(pid:816,state:1)
[ 8233.260395] |irqbalance(pid:817,state:1)
[ 8233.260396] |NetworkManager(pid:819,state:1)---dhclient(pid:999,state:1)
[ 8233.260396] |systemd-logind(pid:820,state:1)
[ 8233.260397] |polkitd(pid:842,state:1)
[ 8233.260398] |unattended-upgr(pid:885,state:1)
[ 8233.260399] |gdm3(pid:909,state:1)---gdm-session-wor(pid:965,state:1)
[ 8233.260400] |---gdm-session-wor(pid:1937,state:1)
[ 8233.260402] |systemd(pid:1108,state:1)---(sd-pam)(pid:1109,state:1)
[ 8233.260402] |---dbus-daemon(pid:1129,state:1)
[ 8233.260403] |---at-spi-bus-laun(pid:1354,state:1)
[ 8233.260403] |---at-spi2-registr(pid:1361,state:1)
[ 8233.260404] |---pulseaudio(pid:1370,state:1)
[ 8233.260404] |---ibus-portal(pid:1473,state:1)
[ 8233.260405] |---xdg-permission-(pid:1486,state:1)
[ 8233.260405] |upowerd(pid:1161,state:1)
[ 8233.260406] |whoopsie(pid:1197,state:1)
[ 8233.260406] |kerneloops(pid:1206,state:1)
[ 8233.260407] |kerneloops(pid:1218,state:1)
[ 8233.260407] |snapd(pid:1237,state:1)
[ 8233.260408] |rtkit-daemon(pid:1371,state:1)
[ 8233.260408] |ibus-x11(pid:1468,state:1)
[ 8233.260409] |boltd(pid:1547,state:1)
[ 8233.260409] |packagekitd(pid:1554,state:1)
```

```
[ 8233.260410] |colord(pid:1644,state:1)
[ 8233.260415] |systemd(pid:1988,state:1)---(sd-pam)(pid:1989,state:1)
[ 8233.260416] |---dbus-daemon(pid:2014,state:1)
[ 8233.260416] |---at-spi-bus-laun(pid:2114,state:1)
[ 8233.260417] |---at-spi2-registr(pid:2121,state:1)
[ 8233.260417] |---gvfsd(pid:2144,state:1)
[ 8233.260418] |---gvfsd-fuse(pid:2149,state:1)
[ 8233.260418] |---ibus-portal(pid:2187,state:1)
[ 8233.260418] |---xdg-permission-(pid:2194,state:1)
[ 8233.260419] |---evolution-sourc(pid:2205,state:1)
[ 8233.260419] |---gnome-shell-cal(pid:2201,state:1)
[ 8233.260420] |---gvfs-udisks2-vo(pid:2225,state:1)
[ 8233.260420] |---gvfs-afc-volume(pid:2229,state:1)
[ 8233.260421] |---gvfs-mtp-volume(pid:2234,state:1)
[ 8233.260421] |---gvfs-gphoto2-vo(pid:2238,state:1)
[ 8233.260422] |---gvfs-goa-volume(pid:2242,state:1)
[ 8233.260422] |---goa-daemon(pid:2213,state:1)
[ 8233.260422] |---goa-identity-se(pid:2251,state:1)
[ 8233.260423] |---nautilus(pid:2390,state:1)
[ 8233.260423] |---evolution-calen(pid:2396,state:1)
[ 8233.260424] |---dconf-service(pid:2427,state:1)
[ 8233.260424] |---evolution-addre(pid:2453,state:1)
[ 8233.260424] |---gvfsd-metadata(pid:2477,state:1)
[ 8233.260425] |---gnome-terminal-(pid:3345,state:1)
[ 8233.260425] |---gnome-keyring-d(pid:2002,state:1)
[ 8233.260426] |---pulseaudio(pid:2160,state:1)
[ 8233.260426] |---ibus-x11(pid:2184,state:1)
[ 8233.260427] |---gsd-printer(pid:2341,state:1)
[ 8233.260427] |---vmttoolsd(pid:2362,state:1)
[ 8233.260428] |---fwupd(pid:2515,state:1)
[ 8233.260428] |---cupsd(pid:3128,state:1)
[ 8233.260429] |---cups-browsed(pid:3129,state:1)
[ 8233.260430] |systemd(pid:3364,state:1)---(sd-pam)(pid:3365,state:1)
root@ubuntu:/home/zhangziyuan/exp2#
```

由上图可知：自己构建的进程树 pstree 输出完全正确。

五 个人实验改进与总结

5.1 个人实验小结

项目实现创新点说明	1、通过阅读源码，明晰了 Linux 系统内核函数的诸多功能与结构； 2、了解了线程和进程之间的关系，学会了线程的表示方法： <code>{name}statistics</code> 。特别地，当 <code>statistics</code> 为 1 的时候，该表示方法 <code>{name}</code> 即进程的含义。
-----------	---

5.2 个人实验总结

实验体会：

通过本次实验，Linux 内核模块编程让我更好地理解 Linux 内核的工作原理。本次实验分为两个步骤：递归查找子进程，以及输出子进程。前者较容易，但后者却苦恼了我很久。

在自主编程的过程中，我用了 C 语言中的一些有意思的机制。宏（定义）函数就是一个有意思的机制，可以有效提高编程效率，增加代码阅读的可读性。由于我对 C 语言基础掌握较为牢固，有很多小功能在实际编程中都很有用，比如：do-while 循环、结构体中的空数组、枚举、联合机制等等。

在这次实验中能发挥这些小部件的大功能，最后不断调整进程树输出格式到完美，成就感爆棚。

六 参考文献

[1] Linux Device Driver (Third Edition). Jonathan Corbet, et al.