

实验概述

[实验环境]

Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-88-generic x86_64)

[实验步骤]

详见Installing OS161.pdf和Working with OS161.pdf

[实验中主要修改的文件]

main.c和menu.c

[实验要求]

详见Assignment.pdf

[要求实验情况]

本实验的要求都已实现。

实验内容

一. Customize the Kernel Boot Output

程序编号	1	文件名	main.c	说明	customize, 只截取部分
<pre>..... kprintf("\n"); kprintf("OS/161 base system version %s\n", BASE_VERSION); kprintf("%s", harvard_copyright); kprintf("\n"); kprintf("WangchuWEN's system version %s (%s # %d)\n", GROUP_VERSION, buildconfig, buildversion); kprintf("\n"); // 思路: 修改 /startup 下的 main.c 文件, 找到输出引导语的部分, 修改 kprintf 中的 内容即可。然后再 bmake, 后把新的 kernel 通过 cp 命令复制到 /root 下。</pre>					

二. Add a Kernel Menu Command

程序编号	2	文件名	menu.c	说明	修改了三个部分
<pre>..... // 修改的第一部分, 用于菜单显示 dth 选项 static const char *opsmenu[] = { "[s] Shell", "[p] Other program", "[mount] Mount a filesystem", "[unmount] Unmount a filesystem", "[bootfs] Set \"boot\" filesystem", "[pf] Print a file", "[cd] Change directory", "[pwd] Print current directory", "[sync] Sync filesystems", "[panic] Intentional panic", "[q] Quit and shut down", "[dth] Print threads' information", NULL };</pre>					

.....

//修改的第二部分，用于调用cmd_showdbthread函数

/* operations */

```
{ "s",          cmd_shell },
{ "p",          cmd_prog },
{ "mount",      cmd_mount },
{ "unmount",    cmd_unmount },
{ "bootfs",     cmd_bootfs },
{ "pf",         printfile },
{ "cd",         cmd_chdir },
{ "pwd",        cmd_pwd },
{ "sync",       cmd_sync },
{ "panic",      cmd_panic },
{ "q",          cmd_quit },
{ "exit",       cmd_quit },
{ "halt",       cmd_quit },
{ "dth",        cmd_showdbthread },
```

.....

.....

//修改的第三部分，用于cmd_showdbthread的具体实现

static

int

cmd_showdbthread(int nargs, char **args)

```
{
    (void)nargs;
    (void)args;

    dbflags = (int)DB_THREADS;

    return 0;
}
```

.....

.....

//思路：修改/startup下的menu.c文件，在菜单中添加dth选项，

cmd_showdbthread函数实现为：将dbflags置为DB_THREADS。然后再bmake，后把新的kernel通过cp命令复制到/root下。

实验结果

[Kernel Boot Output]

```
ubuntu@VM-0-17-ubuntu:~/cs350-os161/root$ sys161 kernel
sys161: System/161 release 1.99.06, compiled Jun 16 2020 15:34:07

OS/161 base system version 1.99.05
Copyright (c) 2000, 2001, 2002, 2003, 2004, 2005, 2008, 2009
  President and Fellows of Harvard College. All rights reserved.

WangchuWEN's system version 0 (ASST0 #4)

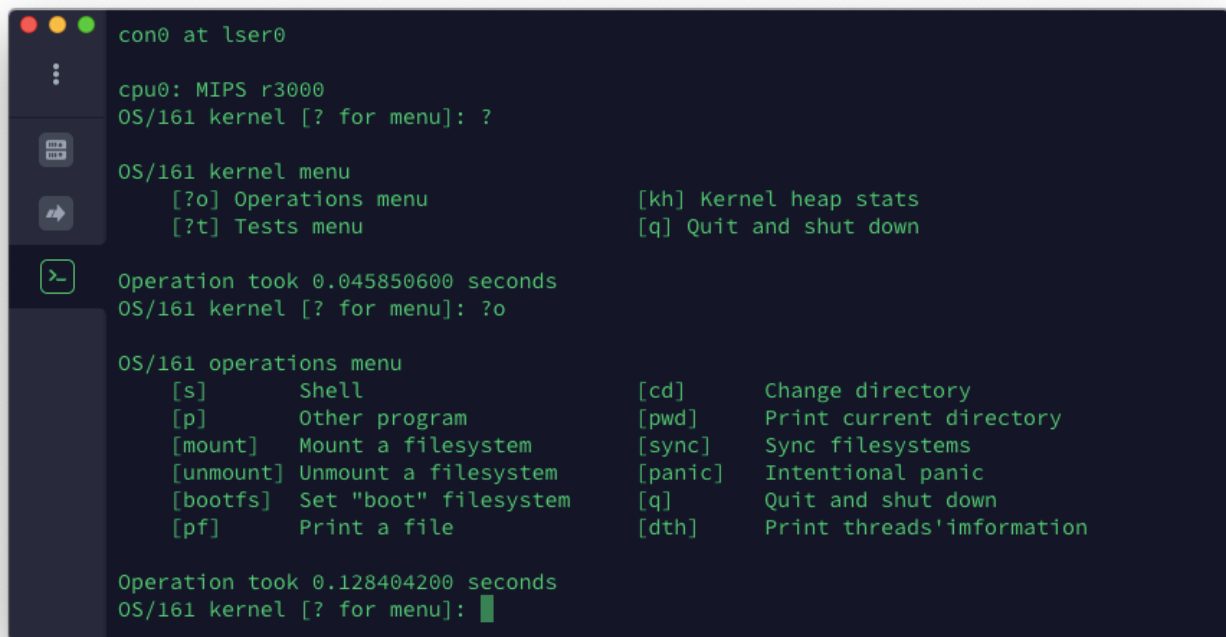
304k physical memory available
Device probe...
lamebus0 (system main bus)
emu0 at lamebus0
ltrac0 at lamebus0
ltime0 at lamebus0
beep0 at ltime0
rtclock0 at ltime0
lrando0 at lamebus0
rando0 at lrando0
lhd0 at lamebus0
lhd1 at lamebus0
lser0 at lamebus0
con0 at lser0

cpu0: MIPS r3000
OS/161 kernel [? for menu]:
```

[Running tt2 without dth/with dth]

```
cpu0: MIPS r3000
OS/161 kernel [? for menu]: tt2
Starting thread test 2...
0123456701235674
Thread test 2 done.
Operation took 0.668436320 seconds
OS/161 kernel [? for menu]: dth
Operation took 0.000020880 seconds
OS/161 kernel [? for menu]: tt2
Starting thread test 2...
Forking thread: threadtest0
F0rking th0read: threadtest1
F1orking th1read: threadtest2
F2orking th2read: threadtest3
F3orking th3read: threadtest4
F4orking th4read: threadtest5
F5orking th5read: threadtest6
F6orking th6read: threadtest7
77
Thread test 2 done.
Operation took 0.723857720 seconds
OS/161 kernel [? for menu]:
```

[MENU]



```
con0 at lser0
cpu0: MIPS r3000
OS/161 kernel [? for menu]: ?

OS/161 kernel menu
  [?o] Operations menu      [kh] Kernel heap stats
  [?t] Tests menu          [q] Quit and shut down

Operation took 0.045850600 seconds
OS/161 kernel [? for menu]: ?o

OS/161 operations menu
  [s]   Shell               [cd]   Change directory
  [p]   Other program       [pwd]  Print current directory
  [mount] Mount a filesystem [sync] Sync filesystems
  [unmount] Unmount a filesystem
  [bootfs] Set "boot" filesystem
  [pf]   Print a file       [panic] Intentional panic
                               [q]    Quit and shut down
                               [dth]  Print threads'information

Operation took 0.128404200 seconds
OS/161 kernel [? for menu]: █
```

遇到的问题 and 解决办法

1.修改了main.c和menu.c后，重新运行kernel，发现修改并没有生效。

解决办法：在仔细阅读了working with OS161.pdf后，清楚了这个系统的运行原理，修改完/startup里的文件后，还需要在/compile/AXXT0中进行bmake，bmake install等操作，并在bmake install的引导语指示下，即：

```
mkdir /home/ubuntu/cs350-os161/root
```

```
cp kernel /home/ubuntu/cs350-os161/root/kernel-ASST0
```

```
rm -f /home/ubuntu/cs350-os161/root/kernel
```

```
ln -s kernel-ASST0 /home/ubuntu/cs350-os161/root/kernel
```

将新编译生成的kernel移到/root目录下，这样就可以通过sys161 kernel来运行新编译的程序了。

2.安装过程中，无法执行source ~/.bashrc保存环境变量，错误提示为：cannot find file。

解决办法，在/etc下找到bash.bashrc文件，并通过vim将export PATH=\$HOME/sys161/bin:\$HOME/sys161/tools/bin:\$PATH写入文件中，sudo保存退出。

3.安装gdb时，出现了和installing OS161.pdf中同样的错误。

解决办法：安装installing OS161.pdf中的指引，依次输入：

```
sudo apt-get install libncurses-dev
```

```
make MAKEINFO=missing
```

```
make install
```

后成功完成安装。

目录结构

```
OS_EXP6/
|-- Instructions/
|   |-- Installing OS161.pdf
|   |-- Working with OS161.pdf
|   |-- Assignment.pdf
|
|-- files modified/
|   |-- main.c
|   |-- menu.c
|
|-- LICENCE
|
|-- README.md
|-- 实验报告.pdf
```

[超链接： 点击此处可在github中查看该项目](#)