

《操作系统原理》实验报告

年级： 2019 级 班级： 大类 8 班 学号： 3019244266 姓名： 李润泽

一、实验题目及要求

实验题目：LAB3-xv6-系统调用

实验要求：1、系统调用添加打印日志功能
2、增加日期系统调用

二、设计说明 (用来说明程序的功能、结构、原理等)

在第一部分中，我们需要修改 syscall.c 中的代码，使其能够为每个系统调用都能输出跟踪信息，跟踪信息由系统调用名称与系统调用返回值组成。

我们在 syscall.c 中加入 sys_name[][10]数组，用于保存系统调用号与系统调用名称的映射关系。

然后在 syscall 函数中加入 `cprintf("%s -> %d\n", sys_name[num], curproc->tf->eax);` 一句来输出系统调用的名字以及该系统调用的返回值。

在第二部分中，我们需要添加一个新的系统调用 date。

首先在 syscall.h 中添加系统调用编号 `#define SYS_date 22`

然后在 syscall.c 文件中添加系统调用函数的外部声明 `extern int sys_date(void);` 等内容。

之后，我们还要在 user.h、usys.S 中添加用户态函数的定义以及实现。

此外，我们还要在 sysproc.c 中添加 sys_date 函数来实现系统调用函数。

最后，新建一个文件 date.c，添加作业文档中老师给予的内容来输出 UTC 时间。

另外，在 Makefile 中需要添加 UPROGS 对应命令的定义 `_date\`

三、编译、运行、测试说明 (简单说明如何编译、运行、测试你提交的代码。如果程序由多个源程序构成，建议编写 Makefile，或者给出编译脚本。)

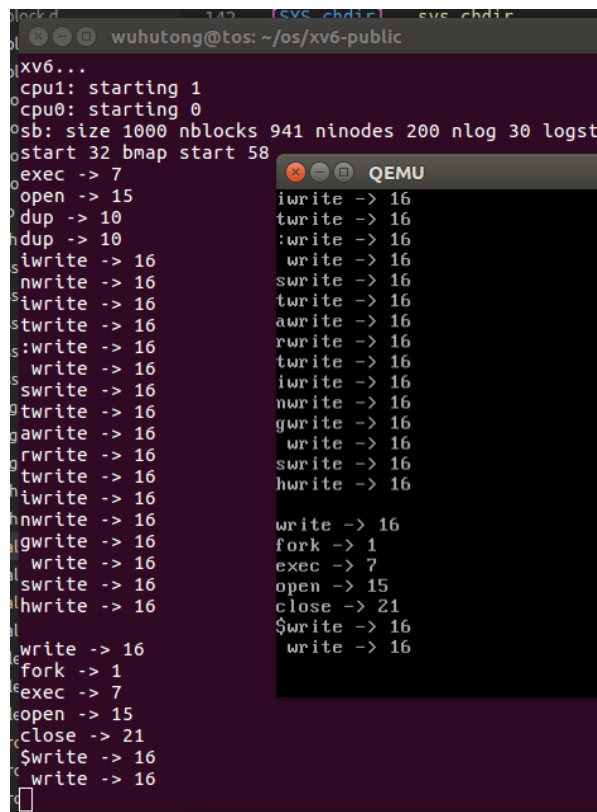
整个实验中，我们需要输入指令 `make qemu` 来验证实验的正确性。

在第二部分中，我们需要首先输入指令 `grep -n uptime *.chs`，这条指令用于查看为了创建 uptime 系统调用了哪些地方的准备工作，这样可以通过“照葫芦画瓢”的方式，为创建 date 系统调用添加相关代码。在 `make qemu` 之后输入 `date` 即可。

```
wuhutong@tos:~/os/xv6-public$ grep -n uptime *.chs
syscall.c:105:extern int sys_uptime(void);
syscall.c:122:[SYS_uptime] sys_uptime,
syscall.c:148:[SYS_uptime] "uptime",
syscall.h:15:#define SYS_uptime 14
sysproc.c:83:sys_uptime(void)
user.h:25:int uptime(void);
wuhutong@tos:~/os/xv6-public$
```

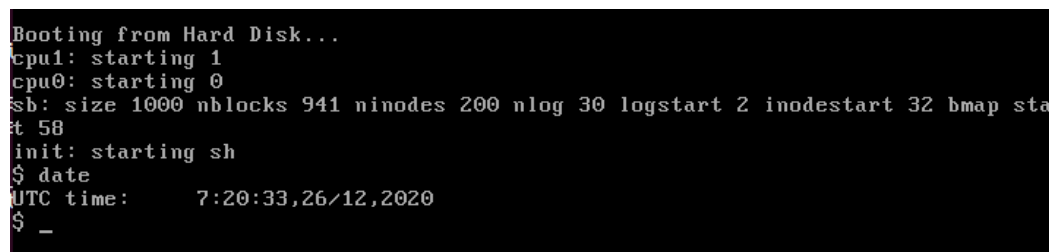
四、实验结果与结论分析（经调试正确的程序的运行结果截图，包括输入数据、输出结果、结论）

第一部分



```
lock d 145 [xv6_chdir] xv6_chdir
wuhutong@tos: ~/os/xv6-public
xv6...
cpu1: starting 1
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logst
start 32 bmap start 58
exec -> 7
open -> 15
dup -> 10
dup -> 10
iwrite -> 16
nwrite -> 16
siwrite -> 16
stwrite -> 16
s:write -> 16
write -> 16
swrite -> 16
twrite -> 16
awrite -> 16
rwrite -> 16
twrite -> 16
iwrite -> 16
nwrite -> 16
gwrite -> 16
write -> 16
swrite -> 16
hwrite -> 16
write -> 16
fork -> 1
exec -> 7
open -> 15
close -> 21
$write -> 16
write -> 16
iwrite -> 16
twrite -> 16
write -> 16
write -> 16
swrite -> 16
rwrite -> 16
twrite -> 16
iwrite -> 16
nwrite -> 16
gwrite -> 16
write -> 16
swrite -> 16
hwrite -> 16
write -> 16
fork -> 1
exec -> 7
open -> 15
close -> 21
$write -> 16
write -> 16
```

第二部分



```
Booting from Hard Disk...
cpu1: starting 1
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58
init: starting sh
$ date
UTC time: 7:20:33,26/12,2020
$ _
```

五、实验中遇到的问题及解决方法

在完成第二部分之前，我发现每次进行 `make qemu` 的时候总会出现第一部分的输出，导致影响时间的显示格式。解决方法为将 `syscall.c` 中的第一部分的代码注释掉，即可输出第二部分的时间。

完成日期: 2020.12.25