

重庆大学

学生实验报告

实验课程名称 操作系统

开课实验室 DS1503

学 院 大数据与软件学院 年级 软件工程 专业班
01

学 生 姓 名 邓永思 学 号 20231265

开 课 时 间 2024 至 2025 学年第 二 学期

总 成 绩	
教师签名	

《操作系统》实验报告

开课实验室：

2025 年 3 月 1 日

学院	大数据与软件学院	年级、专业、班	2023/软件工程 /01	姓名	邓永思	成绩	
课程名称	操作系统	实验项目名称	实验一：系统调用		指导教师	刘寄	
教师评语	<p style="text-align: right;">教师签名： 年 月 日</p>						

一、实验目的

安装实验环境，尝试自定义系统调用，验证两种路径的输出是否一致

二、实验内容

1.配置实验环境并尝试简单命令

检出（checkout）EPOS 的源代码

```
svn checkout https://svn.riouxsvn.com/epos
```

注意：不要从 GitHub 克隆！

编译及运行

```
cd epos
```

```
make run
```

清除所有的临时文件

```
make clean
```

2.编写系统调用 “time_t time(time_t *loc)”

功能描述

返回从格林尼治时间 1970 年 1 月 1 日午夜起所经过的秒数。如果指针 loc 非 NULL，则返回值也被填到 loc 所指向的内存位置

三、使用仪器、材料

虚拟机，编译器

三、实验过程原始记录(数据、图表、计算等)

函数实现

```
    }
    time_t sys_time()
    {
        time_t uptime_seconds = g_timer_ticks / HZ; //计算自启动以来的时间
        time_t current_time = g_startup_time + uptime_seconds; //计算当前时间戳
        return current_time;
    }
```

```
task_getid(), pv);

//TODO: Your code goes here
time_t t1, t2;
t1 = time(&t2);
printf("t1= %ld\n", t1);
printf("t2= %ld\n", t2);

while(1)
```

部分内核与用户态的环境配置

```
#define SYSCALL_putchar      1000
#define SYSCALL_getchar      1001
#define SYSCALL_time         2025
```

```
#endif /*_SYSCALLNR_H*/
```

```
void mi_startup();
#endif /*_KERNEL_H*/
time_t sys_time();
```

```
break;
case SYSCALL_time:
{
    time_t* loc = *(time_t**) (ctx->esp + 4);
    ctx->eax = sys_time();
    if (loc != NULL)
        *loc = ctx->eax;
    break;
}
```

```
#endif /*_SYSCALL_H*/
time_t time(time_t* loc);
```

五、实验结果及分析

1. 熟悉环境

```
epos\userapp\include\math.h
epos\userapp\include\setjmp.h
epos\userapp\include\syscall.h
epos\userapp\lib\malloc.c
epos\userapp\lib\setjmp.S
epos\userapp\lib\syscall-wrapper.S
epos\userapp\graphics.c
epos\userapp\myalloc.c
epos\Makefile
epos\userapp\include\stdlib.h
epos\userapp\lib\crt0.S
epos\userapp\lib\qsort.c
epos\userapp\lib\stdlib.c
epos\userapp\include\stdio.h
epos\userapp\include\unistd.h
epos\userapp\lib\math.c
epos\userapp\lib\stdio.c
epos\userapp\lib\sysconf.c
epos\userapp\graphics.h
epos\userapp\vm86call.c
epos\Makefile.inc
epos\userapp\Makefile
epos\userapp\main.c
epos\.gdbinit
epos\hd.img.bz2.txt
checked out revision 96.
```



```
trial (正文) v | 24
QEMU
Machine View
Booting EPOS
Filesystem type is [Multiboot-kludge]
y@0x1000000
Welcome to EPOS
Copyright (C) 2005-2006
All rights reserved.
RAM: 0x00002000 - 0x00128000
Calibrating delay...
task #0: Initializin
task #0: Initializin
task #0: Initializin
task #0: Loading a d
task #0: Creating fi
t80!
t1= 1740793886
t2= 1740793886
make[1]: Leaving directory '/d/2025_02_27_expenv/expenv/epos/userapp'
if [ ! -s hd.img ]; then base64 -d hd.img.bz2.txt | bunzip2 >hd.img ; fi
imgcp kernel/eposkrnl.bin hd.img=C:\eposkrnl.bin
imgcp userapp/a.out hd.img=C:\a.out
qemu-system-i386w -m 32 -boot order=c -vga std -drive format=raw,file=hd.img,index=0,media=disk
```

2. 关于时间的系统调用，T1与t2的值一致，符合实验预期

```
RAM: 0x00002000 - 0x0009f000 (157 frames)
RAM: 0x00128000 - 0x01fe0000 (7864 frames)
Calibrating delay... 128614400 loops per second (257.22 Bo
task #0: Initializing IDE controller...Done
task #0: Initializing PCI controller...Done
task #0: Initializing FAT file system...Done
task #0: Loading a.out...Done
task #0: Creating first user task...task #1: I'm the first
78) !
t1= 1740791932
t2= 1740791932
```

实验报告打印格式说明

1. 标题：三号加粗黑体
2. 开课实验室：5号加粗宋体
3. 表中内容：
 - (1) 标题：5号黑体
 - (2) 正文：5号宋体
4. 纸张：16开(20cm×26.5cm)
5. 版芯
上距：2cm

下距: 2cm

左距: 2.8cm

右距: 2.8cm

说明: 1、“年级专业班”可填写为“00 电子 1 班”，表示 2000 级电子工程专业第 1 班。

2、实验成绩可按五级记分制（即优、良、中、及格、不及格），或者百分制记载，若需要将实验成绩加入对应课程总成绩的，则五级记分应转换为百分制。