

# 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          学 号         

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		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	<div>教师签名：</div> <div>年   月   日</div>						

## 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

```

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

```

```
gcc -m32 -march=i386 -Wall -pipe -DSNPRINTF_FLOATPOINT -ffreestanding -fleading-underscore -mno-ms-b  
s -fno-pie -fno-stack-check -mno-stack-arg-probe -fno-stack-protector -O2 -ggdb -nostdinc -I../includ  
clude -c -o ../lib/string.o ../lib/string.c  
gcc -m32 -march=i386 -Wall -pipe -DSNPRINTF_FLOATPOINT -ffreestanding -fleading-underscore -mno-ms-b  
s -fno-pie -fno-stack-check -mno-stack-arg-probe -fno-stack-protector -O2 -ggdb -nostdinc -I../includ  
clue -c -o ../lib/memcpy.o ../lib/memcpy.c  
gcc -m32 -march=i386 -Wall -pipe -DSNPRINTF_FLOATPOINT -ffreestanding -fleading-underscore -mno-ms-b  
s -fno-pie -fno-stack-check -mno-stack-arg-probe -fno-stack-protector -O2 -ggdb -nostdinc -I../includ  
clue -c -o ../lib/memset.o ../lib/memset.c  
gcc -m32 -march=i386 -Wall -pipe -DSNPRINTF_FLOATPOINT -ffreestanding -fleading-underscore -mno-ms-b  
s -fno-pie -fno-stack-check -mno-stack-arg-probe -fno-stack-protector -O2 -ggdb -nostdinc -I../includ  
clue -c -o ../lib/sprintf.o ../lib/sprintf.c  
gcc -m32 -march=i386 -Wall -pipe -DSNPRINTF_FLOATPOINT -ffreestanding -fleading-underscore -mno-ms-b  
s -fno-pie -fno-stack-check -mno-stack-arg-probe -fno-stack-protector -O2 -ggdb -nostdinc -I../includ  
clue -c -o lib/malloc.o lib/malloc.c  
gcc -m32 -march=i386 -Wall -pipe -DSNPRINTF_FLOATPOINT -ffreestanding -fleading-underscore -mno-ms-b  
s -fno-pie -fno-stack-check -mno-stack-arg-probe -fno-stack-protector -O2 -ggdb -nostdinc -I../includ  
clue -c -o ../lib/tlsf/tlsf.o ../lib/tlsf/tlsf.c  
gcc -m32 -nostdlib -nostartfiles -nodefaultlibs -Wl,-Map,a.map -static -o a.out lib/crt0.o lib/stdio.o lib  
/syscalls-wrapper.o vm86call.o graphics.o main.o lib/sysconf.o lib/math.o lib/stdio.o lib/stdlib.o lib  
/o ../lib/softfloat.o ../lib/string.o ../lib/memcpy.o ../lib/memset.o ../lib/sprintf.o lib/malloc.o  
/tlsf/tlsf.o  
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  
imgcpy kernel/eposkrnl.bin hd.img:C:\eposkrnl.bin  
imgcpy 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。