

# 同济大学计算机系

## 操作系统实验报告



实验内容 在 UNIX V6++ 中添加新的系统调用

学 号 2251745

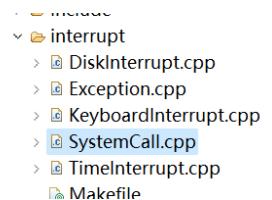
姓 名 张宇

专 业 计算机科学与技术

授课老师 方钰

# 一、在 UNIX V6++中添加一个新的系统调用接口

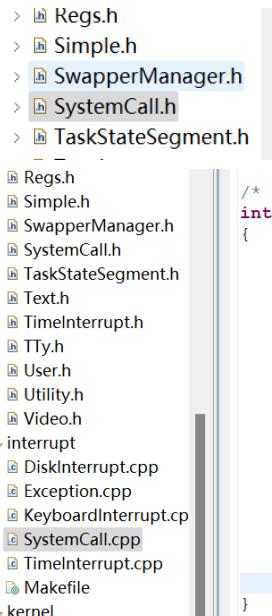
## 1.1 在系统调用处理子程序入口表中添加新的入口



```
/* 48 = sig */
/* 49 = getppid */
/* 50 = nosys */
/* 51 = nosys */
/* 52 = nosys */
/* 53 = nosys */
/* 54 = nosys */
/* 55 = nosys */

{ 2, &Sys_Ssig },
{ 1, &Sys_Getpid},
{ 0, &Sys_Nosys },
```

## 1.2 在 SYSTEMCALL 类中添加新的系统调用处理子程序



```
/* 49 = getppid count = 1 */
static int Sys_Getpid();

/* 50 ~ 63 = nosys count = 0 */

/* 49 = getppid count = 1 */
int SystemCall::Sys_Getpid()
{
    ProcessManager& procMgr = Kernel::Instance().GetProcessManager();
    User& u = Kernel::Instance().GetUser();

    int i;
    int curpid = (int)u.u_arg[0];

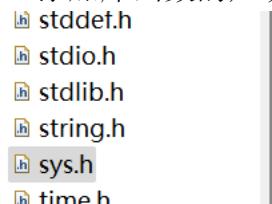
    u.u_ar0[User::EAX] = -1;

    for (i = 0; i < ProcessManager::NPROC; i++)
    {
        if (procMgr.process[i].p_pid == curpid)
        {
            u.u_ar0[User::EAX] = procMgr.process[i].p_ppid;
        }
    }

    return 0;
}
```

# 二、为新的系统调用添加新的库函数

## 2.1 添加库函数的声明

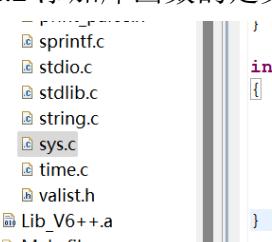


```
int getPath(char* path);

int getpid();

unsigned int getgid();
```

## 2.2 添加库函数的定义



```
int getppid(int pid)
{
    int res;
    __asm__ volatile ("int $0x80": "=a"(res): "a"(49), "b"(pid));
    if (res >= 0)
        return res;
    return -1;
}
```

## 2.3 进行编译

CDI Build Console [oos]

```

> proc
> program
> shell
> test
> tty
  LinkId
  Makefile
  Makefile.inc
> targets
> tools

已复制 1 个文件。
copy ..\targets\objs\kernel.bin ..\tools\MakeImage\bin\Debug\kernel.bin
已复制 1 个文件。
copy ..\targets\img\c.img ..\tools\MakeImage\bin\Debug\c.img
已复制 1 个文件。
cd ..\tools\MakeImage\bin\Debug && build.exe c.img boot.bin kernel.bin
programs
copy ..\tools\MakeImage\bin\Debug\c.img "..\targets\UNIXV6++"\c.img
已复制 1 个文件。

**** Build Finished ****

```

### 三、添加测试程序进行测试

#### 3.1 添加代码并编译生成程序

File Explorer:

- > machine
- > mm
- > pe
- > proc
- > program
  - > objs
    - cat.c
    - cat1.c
    - copyfile.c
    - cp.c
    - date.c
    - echo.c
    - forks.c
    - GetOptAndPath.h
    - getppid.c
    - ls.c
- > newsig.c
- > performance.c
- > rm.c
- > showStack.c
- > shutdown.c
- > newsig.c
- > performance.c
- > rm.c
- > showStack.c
- > shutdown.c

Code Editor: SystemCall.h

```
#include <stdio.h>
#include <sys.h>

int main()
{
    int pid, ppid;
    pid = getpid();
    ppid = getppid(pid);

    printf("This is Process %d# speaking...\n", pid);
    printf("My parent process ID is: %d\n", ppid);

    return 0;
}
```

Build Log:

```
$ (TARGET)\getppid.exe : getppid.c
$ (CC) $ (CFLAGS) -I"$ (INCLUDE)" -I"$ (LIB_INCLUDE)" $ <-e _main1 $ (V6++LIB) -o $ @
copy $ (TARGET)\getppid.exe $ (MAKEIMAGEPATH)\$ (BIN)\getppid.exe

clean:
del c:\temp\cdt\*.*

$ (TARGET)\getppid.exe : getppid.c
$ (CC) $ (CFLAGS) -I"$ (INCLUDE)" -I"$ (LIB_INCLUDE)" $ <-e _main1 $ (V6++LIB) -o $ @
copy $ (TARGET)\getppid.exe $ (MAKEIMAGEPATH)\$ (BIN)\getppid.exe

clean:
del c:\temp\cdt\*.*
```

#### 3.2 运行和调试

File Explorer:

- > proc
- > program
- > shell
- > test
- > tty

Code Editor: SystemCall.cpp

```
int SystemCall::Sys_Getppid()
{
    ProcessManager& procMgr = Kernel::Instance().GetProcessManager();
    User& u = Kernel::Instance().GetUser();

    int i;
    int curpid = (int)u.u_arg[0];
    u.u_ar0[User::EAX] = -1;

    for (i = 0; i < ProcessManager::NPROC; i++)
    {
        if (procMgr.process[i].p_pid == curpid)
        {
            u.u_ar0[User::EAX] = procMgr.process[i].p_ppid;
        }
    }

    return 0;
}
```

Registers:

u_ssav	0xc03ff008
u_procp	<incomplete type>
u_MemoryDescriptor	{...}
u_ar0	0xc03ffd0c
*u_ar0	49
u_arg	0xc03ff030
u_dirp	<incomplete type>
u_utime	0

Registers:

u_ssav	0xc03ff008
u_procp	<incomplete type>
u_MemoryDescriptor	{...}
u_ar0	0xc03ffd0c
*u_ar0	1
u_arg	0xc03ff030
u_dirp	<incomplete type>
u_utime	0

	C03FFFFA0	00000002	C03FFFFB0	00000000	00000000	00000023
◆	0xc011ae94		C03FFFFB0	C03FFFEC	00000000	00000000
◆	0xc0208000		C03FFFC0	000000023	00000002	00000001
◆	0xc0200000		C03FFFD0	000E0000	0000FFAC	C03FFE8
◆	0xc0201000		C03FFFE0	00000016	C03FFFEC	007FFFB8
◆	0xc0202000		C03FFF0	0000001B	00000216	007FFFAC
◆	0xc0203000		C0400000	000E0000	0000FFAC	C03FFE8
◆	0xc03ffffdc		C0400010	000E0000	0000FFAC	C03FFE8
	-----	-----	-----	-----	-----	-----

```
[/]#cd bin  
[/bin]#getppid.exe  
This is Process 2# speaking...  
My parent process ID is: 1  
[/bin]#_
```