

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

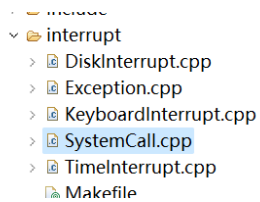
姓 名 张宇

专 业 计算机科学与技术

授课老师 方钰

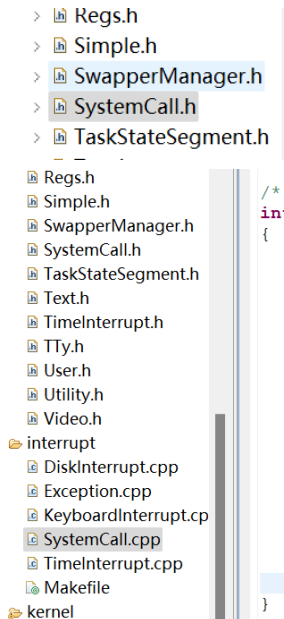
一、在 UNIX V6++中添加一个新的系统调用接口

1.1 在系统调用处理子程序入口表中添加新的入口



```
{ 2, &Sys_Ssig }, /* 48 = sig */
{ 1, &Sys_Getppid }, /* 49 = getppid */
{ 0, &Sys_Nosys }, /* 50 = nosys */
{ 0, &Sys_Nosys }, /* 51 = nosys */
{ 0, &Sys_Nosys }, /* 52 = nosys */
{ 0, &Sys_Nosys }, /* 53 = nosys */
{ 0, &Sys_Nosys }, /* 54 = nosys */
{ 0, &Sys_Nosys }, /* 55 = 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_Getppid();

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

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

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

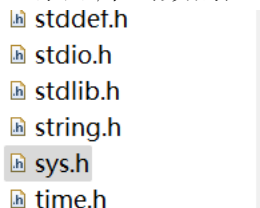
    u.u_ar0[User::EAX] = -1;

    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;
}
```

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

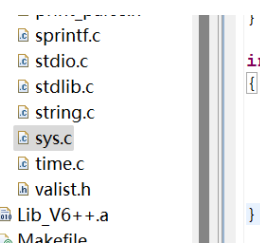


```
int getPath(char* path);

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

- proc
- program
- shell
- test
- tty
- Link.ld
- Makefile
- Makefile.inc
- targets
- tools

CD Build Console [oos]

```

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

- 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

SystemCall.h sys.h sys.c MemoryDescriptor.h getppid.c Process.cpp

```

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

int main1()
{
    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;
}

```

```

$(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 $(TARGET)\* ~

$(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 $(TARGET)\* ~

```

3.2 运行和调试

- proc
- program
- shell
- test
- ttv

```

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

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

for (i = 0; i < ProcessManager::NPROC; i++)

```

```

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;

    {
        if (procMgr.process[i].p_pid == curpid)
        {
            u.u_ar0[User::EAX] = procMgr.process[i].p_ppid;
        }
    }

    return 0;
}

```

u_ssav	0xc03ff008
u_procp	<incomplete type>
u_MemoryDescriptor	{...}
u_ar0	0xc03fffdc
*u_ar0	49
u_arg	0xc03ff030
u_dirp	<incomplete type>
u_ftime	0

u_ssav	0xc03ff008
u_procp	<incomplete type>
u_MemoryDescriptor	{...}
u_ar0	0xc03fffdc
*u_ar0	1
u_arg	0xc03ff030
u_dirp	<incomplete type>
u_ftime	0

- ◆ 0xc011ae94
- ◆ 0xc0208000
- ◆ 0xc0200000
- ◆ 0xc0201000
- ◆ 0xc0202000
- ◆ 0xc0203000
- ◆ 0xc03fffdc

C03FFFB0	C03FFFE0	00000000	00000000	00000023
C03FFFC0	00000023	00000002	00000001	00407030
C03FFFD0	000E0000	0000FFAC	C03FFFE8	00000031
C03FFFE0	00000016	C03FFFE0	007FFFB8	0040141C
C03FFFF0	0000001B	00000216	007FFFAC	00000023
C0400000	000E0000	0000FFAC	C03FFFE8	00000031
C0400010	000E0000	0000FFAC	C03FFFE8	00000031

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