**同济大学计算机系**

**操作系统P04实验报告**

****

**学 号 2151769**

**姓 名 吕博文**

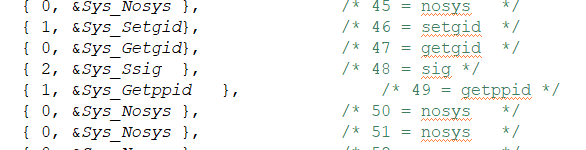
**专 业 计算机科学与技术**

**授课老师 邓蓉**

实验四：UNIX V6++中添加新的系统调用

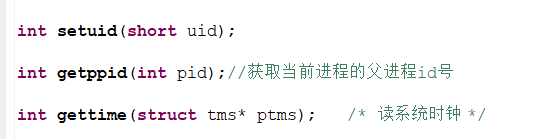
1. **完成实验 4.1，截图说明操作过程，掌握在 UNIX V6++中添加一个新的 系统调用的方法，并总结出主要步骤**
2. **在系统调用子程序入口表中添加新的入口**

如实验文档所示，我们在SystemCall.cpp中找到对系统调用子程序入口表m\_SystemEntranceTable赋值的一段代码并选择第49项设置为getppid项



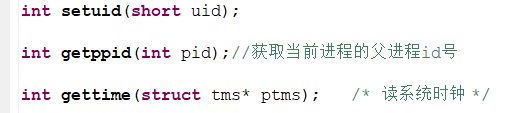
这里加入的子程序名位Sys\_Getppid,后续实现也相应的实现了得到当前进程父进程id号的功能

1. **在SystemCall类中添加系统调用子程序的定义**

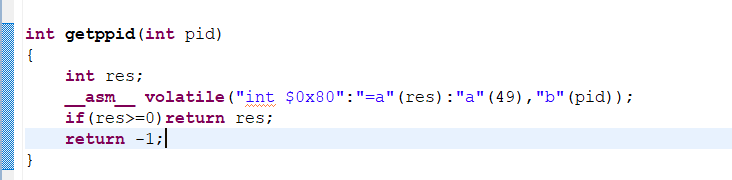
在SystemCall.h文件中添加该系统调用处理子程序的声明如下： 其次，在SystemCall.cpp中添加对应的实现



1. 总结：在UNIX V6++中添加一个新的系统调用步骤为：（1）在系统调用子程序入口表中添加新的入口；（2）在SystemCall类中添加系统调用子程序的定义
2. **完成实验 4.2，掌握在 UNIX V6++中添加库函数的方法，截图说明主要 操作步骤。**
3. 在src/lib/src/sys.h中添加库函数的声明：



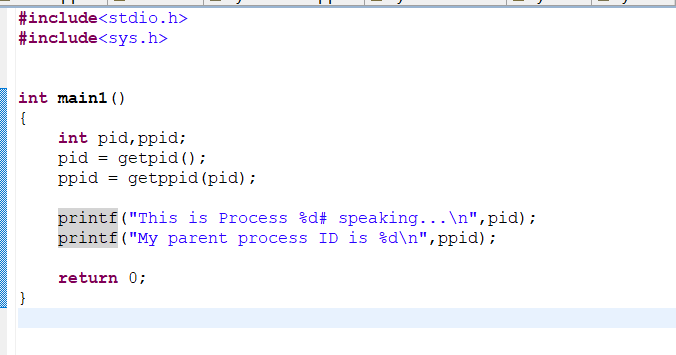
1. 在sys.c中添加库函数的定义



1. 总结，在UNIX V6++中添加库函数的方法步骤为：在src/lib/src/sys.h中添加库函数的声明；在sys.c中添加库函数的定义

**三、完成实验 4.3 ~ 4.4，编写测试程序，通过调试运行说明添加的系统调用的 正确，截图说明主要的调试过程和关键结果**。

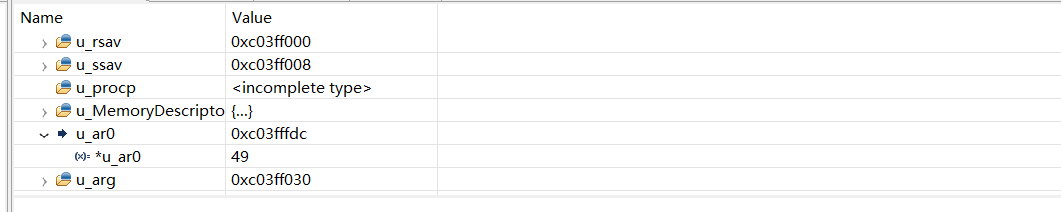
（1）首先编写简单测试程序，这里直接用之前的showStack.exe作为测试程序，改动了程序中的代码实现:



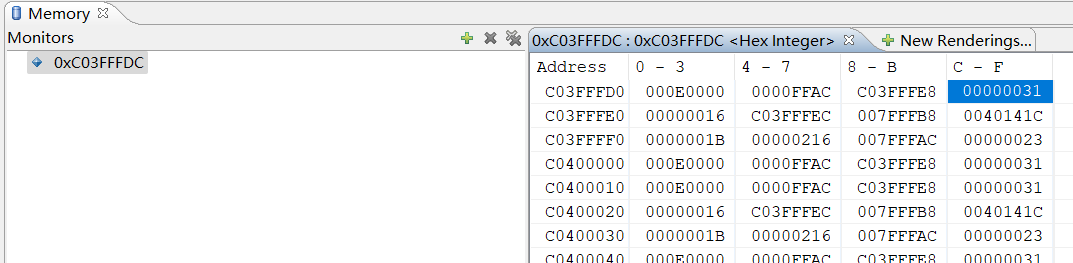
（2）在运行模式下进入UNIX V6++运行测试程序，观察测试结果如下：



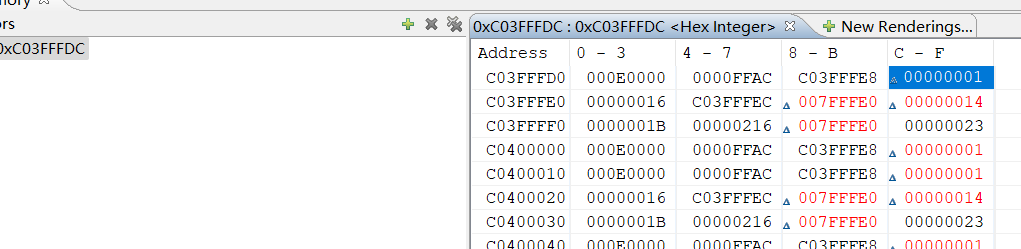
（3）因为涉及到到核心态函数的运行调试，我们设置调试对象为Kernel.exe，在调试模式下进行调试如下：



观察到u\_ar0的值为49



当前内存中观察也是十六进制的0x31为10进制的49

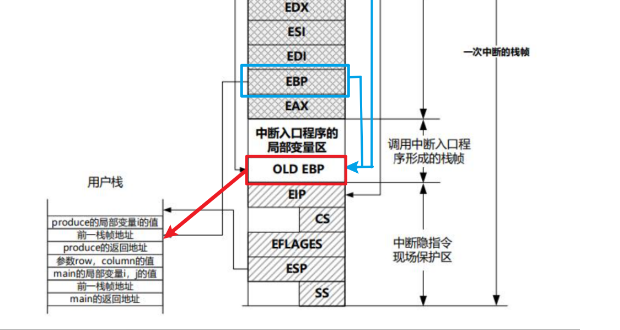


调试运行结束后，我们发现，u\_ar0存储的值变为1，也和最后屏幕上输出的结果当前进程的父进程为1号进程相吻合。

**四、在完成 4.4 的基础上，设计调试方案，确定图 10 中黄色标注的几个地址 单元分别是什么。**

我们将断点设置在getppid函数的汇编语言执行前，首先，我们在本地运行出的ESP值为0x007ffac,EIP值为0x0040141c,EIP上方值为0x007fffb8,最上方的EBP值为0xC03FFFE8;

我们可以考虑使用以下图来理解四个地址的含义：



最下方的ESP表示原来用户栈中的ESP值，之后的EIP表示getppid函数接下来要执行的语句地址，在本例中及是if(res<=0)return -1语句，之后的0x007fffb8是OLD EBP保存用户栈的EBP，最上方的EBP保存当前getppid函数的EBP的值。