

解读linux内核源码的入门方法(下) PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022__E8_A7_A3_E8_AF_BBlinu_c103_144263.htm 【三】对内核源码的修改：1.

在kernel/sys.c中增加系统服务例程如下：
asmlinkage int sys_addtotal(int numdata) { int i=0, enddata=0. while(i < enddata =i . return enddata. } 该函数有一个 int 型入口参数 numdata，并返回从 0 到 numdata 的累加值。当然也可以把系统服务例程放在一个自己定义的文件或其他文件中，只是要在相应文件中作必要的说明；
2.把 asmlinkage int sys_addtotal(int) 的入口地址加到sys_call_table表中: arch/i386/kernel/entry.S 中的最后几行源代码修改前为: _NAME(sys_sendfile) .long

```
SYMBOL_NAME(sys_ni_syscall) /* streams1 */ .long
```

```
SYMBOL_NAME(sys_ni_syscall) /* streams2 */ .long
```

```
SYMBOL_NAME(sys_vfork) /* 190 */ .rept NR_syscalls-190 .long
```

```
SYMBOL_NAME(sys_ni_syscall) .endr 修改后为: ... .. .long
```

```
SYMBOL_NAME(sys_sendfile) .long
```

```
SYMBOL_NAME(sys_ni_syscall) /* streams1 */ .long
```

```
SYMBOL_NAME(sys_ni_syscall) /* streams2 */ .long
```

```
SYMBOL_NAME(sys_vfork) /* 190 */ /* add by I */ .long
```

```
SYMBOL_NAME(sys_addtotal) .rept NR_syscalls-191 .long
```

```
SYMBOL_NAME(sys_ni_syscall) .endr 3. 把增加的 sys_call_table
```

表项所对应的向量,在include/asm-386/unistd.h 中进行必要申明,以供用户进程和其他系统进程查询或调用: 增加后的部分

/usr/src/linux/include/asm-386/unistd.h 文件如下: #define

```
__NR_sendfile 187 #define __NR_getpmsg 188 #define
```

```
__NR_putpmsg 189 #define __NR_vfork 190 /* add by I */ #define
__NR_addtotal 191 4.测试程序(test.c)如下: #include #include
_syscall1(int,addtotal,int, num) main() { int i,j. do printf("Please
input a number\n"). while(scanf("%d",&i)==EOF).
if((j=addtotal(i))== -1) printf("Error occurred in
syscall-addtotal().\n"). printf("Total from 0 to %d is %d \n",i,j). }
```

100Test 下载频道开通 , 各类考试题目直接下载。详细请访问
www.100test.com