

操作系统课程设计实验报告

一、实验内容：

创建 `int sys_wolfie` 系统调用，用于它将 ASCII 图像复制到用户提供的缓冲区，前提是缓冲区足够大。如果缓冲区太小，或无效，返回负值。如果调用成功，返回复制的字节数。

编写一个名为 `wolfietest` 的用户级应用程序。c，从内核获取 `Wolfie` 映像，并将其打印到控制台。

二、实验步骤：

1. 修改 `syscall.h`，添加：

```
#define SYS_wolfie 22
```

2. 修改 `syscall.c`，该文件包含一个函数指针数组，它使用上面定义的索引作为指针，指向在不同位置定义的系统调用，添加：

```
[SYS_wolfie] sys_wolfie,
```

3. 修改 `sysproc.c`，添加：

```
int
```

```
sys_wolfie(void)
```

```
{
```

```
void* buf;
```

```
uint size;
```

```
argptr(0, (void*)&buf, sizeof(buf));
```

```
argptr(1, (void*)&size, sizeof(size));
```

```
char text[] = " Hello! \n\
```

```
    ^ / | \n\
```

```
    / \ 7 < _ / \n\
```

```
    / | / / \n\
```

```
    | z _ , < / / ` \ \n\
```

```
    | \ / > \n\
```

```
    Y ` // \n\
```

```
    ı ● 、 ● ⊂ ⊃ < / \n\
```

```
    () ^ | \ < \n\
```

```
    > - \ _ ı | / / \n\
```

```
    / ^ / / < | \ \ \n\
```

```
    \ _ / ( _ / | / / \n\
```

```
    7 | / \n\
```

```
    > - r _____ ` - - _ \n";
```

```

    if(sizeof(text)>size)

        return -1;

    strncpy((char *)buf, text, size);

    return sizeof(text);

}

```

4. 修改 `usys.S` 和 `user.h` 添加用户程序调用系统调用的接口，分别添加：

```

SYSCALL(wolfie)
int wolfie(void*, uint);

```

5. 创建用户程序 `wolfietest.c`:

```

#include "types.h"
#include "stat.h"
#include "user.h"

int
main(void)
{
    static char buf[2000];
    printf(1, "wolfie sys call returns %d\n", wolfie((void*) buf, 2000));

    printf(1, "%s", buf);
    exit();
}

```

6. 修改 `Makefile` 并重新编译。

三、实验结果：

