xv6-OS-Lab3-syscall

需要修改的文件

```
[*]syscall.h
[*]syscall.c
[*]usys.S
[*]user.h
[*]sysproc.c
[*]Makefile
[+]wolfietest.c
```

接下来一一解释每个文件的定义

文件定义

syscall.h

定义了系统调用的编号

syscall.c

定义系统调用的指针

usys.S

系统调用的汇编文件

user.h

定义用户程序调用的函数

sysproc.c

定义系统调用函数（`wolfie`在这里写）

wolfietest.c

用户程序

Makefile

Makefile在`fs.img`的目标中用`mkfs`生成了系统调用文件，需要在`UPROGS`和`EXTRA`中添加`wolfietest`

实验过程

1. 在`syscall.h`中添加系统调用编号

```
// System call numbers
#define SYS_fork    1
#define SYS_exit    2
#define SYS_wait    3
#define SYS_pipe    4
#define SYS_read    5
#define SYS_kill    6
#define SYS_exec    7
#define SYS_fstat   8
#define SYS_chdir   9
#define SYS_dup    10
#define SYS_getpid 11
#define SYS_sbrk   12
#define SYS_sleep  13
#define SYS_uptime 14
#define SYS_open   15
#define SYS_write  16
#define SYS_mknod  17
#define SYS_unlink 18
#define SYS_link   19
#define SYS_mkdir  20
#define SYS_close  21
#define SYS_wolfie 22 // <- Lab-3
```

2. 在`syscall.c`中添加函数原型和指向系统调用的指针

```
#include "types.h"
#include "defs.h"
#include "param.h"
#include "memlayout.h"
#include "mmu.h"
#include "proc.h"
#include "x86.h"
#include "syscall.h"

// User code makes a system call with INT T_SYSCALL.
// System call number in %eax.
// Arguments on the stack, from the user call to the C
// library system call function. The saved user %esp points
// to a saved program counter, and then the first argument.

// Fetch the int at addr from the current process.
int
fetchint(uint addr, int *ip)
{
  struct proc *curproc = myproc();

  if(addr >= curproc->sz || addr+4 > curproc->sz)
    return -1;
  *ip = *(int*)(addr);
  return 0;
```

```
}

// Fetch the nul-terminated string at addr from the current process.
// Doesn't actually copy the string - just sets *pp to point at it.
// Returns length of string, not including nul.
int
fetchstr(uint addr, char **pp)
{
  char *s, *ep;
  struct proc *curproc = myproc();

  if(addr >= curproc->sz)
    return -1;
  *pp = (char*)addr;
  ep = (char*)curproc->sz;
  for(s = *pp; s < ep; s++){
    if(*s == 0)
      return s - *pp;
  }
  return -1;
}

// Fetch the nth 32-bit system call argument.
int
argint(int n, int *ip)
{
  return fetchint((myproc()->tf->esp) + 4 + 4*n, ip);
}

// Fetch the nth word-sized system call argument as a pointer
// to a block of memory of size bytes.  Check that the pointer
// lies within the process address space.
int
argptr(int n, char **pp, int size)
{
  int i;
  struct proc *curproc = myproc();

  if(argint(n, &i) < 0)
    return -1;
  if(size < 0 || (uint)i >= curproc->sz || (uint)i+size > curproc->sz)
    return -1;
  *pp = (char*)i;
  return 0;
}

// Fetch the nth word-sized system call argument as a string pointer.
// Check that the pointer is valid and the string is nul-terminated.
// (There is no shared writable memory, so the string can't change
// between this check and being used by the kernel.)
int
argstr(int n, char **pp)
{
  int addr;
  if(argint(n, &addr) < 0)
    return -1;
  return fetchstr(addr, pp);
}

extern int sys_chdir(void);
extern int sys_close(void);
extern int sys_dup(void);
extern int sys_exec(void);
extern int sys_exit(void);
extern int sys_fork(void);
extern int sys_fstat(void);
extern int sys_getpid(void);
extern int sys_kill(void);
extern int sys_link(void);
extern int sys_mkdir(void);
extern int sys_mknod(void);
extern int sys_open(void);
extern int sys_pipe(void);
extern int sys_read(void);
extern int sys_sbrk(void);
extern int sys_sleep(void);
extern int sys_unlink(void);
extern int sys_wait(void);
extern int sys_write(void);
extern int sys_uptime(void);
extern int sys_wolfie(void); // <- Lab-3

static int (*syscalls[])(void) = {
[SYS_fork]    sys_fork,
[SYS_exit]    sys_exit,
[SYS_wait]    sys_wait,
[SYS_pipe]    sys_pipe,
[SYS_read]    sys_read,
[SYS_kill]    sys_kill,
[SYS_exec]    sys_exec,
[SYS_fstat]   sys_fstat,
[SYS_chdir]   sys_chdir,
[SYS_dup]     sys_dup,
[SYS_getpid]  sys_getpid,
[SYS_sbrk]    sys_sbrk,
[SYS_sleep]   sys_sleep,
[SYS_uptime]  sys_uptime,
[SYS_open]    sys_open,
[SYS_write]   sys_write,
[SYS_mknod]   sys_mknod,
[SYS_unlink]  sys_unlink,
[SYS_link]    sys_link,
[SYS_mkdir]   sys_mkdir,
[SYS_close]   sys_close,
[SYS_wolfie]  sys_wolfie, // <- Lab-3
};

void
```

```
syscall(void)
{
  int num;
  struct proc *curproc = myproc();

  num = curproc->tf->eax;
  if(num > 0 && num < NELEM(syscalls) && syscalls[num]) {
    curproc->tf->eax = syscalls[num]();
  } else {
    cprintf("%d %s: unknown sys call %d\n",
            curproc->pid, curproc->name, num);
    curproc->tf->eax = -1;
  }
}
```

3. 在`usys.S`中添加汇编

```
#include "syscall.h"
#include "traps.h"

#define SYSCALL(name) \
  .globl name; \
  name: \
    movl $SYS_ ## name, %eax; \
    int $T_SYSCALL; \
    ret

SYSCALL(fork)
SYSCALL(exit)
SYSCALL(wait)
SYSCALL(pipe)
SYSCALL(read)
SYSCALL(write)
SYSCALL(close)
SYSCALL(kill)
SYSCALL(exec)
SYSCALL(open)
SYSCALL(mknod)
SYSCALL(unlink)
SYSCALL(fstat)
SYSCALL(link)
SYSCALL(mkdir)
SYSCALL(chdir)
SYSCALL(dup)
SYSCALL(getpid)
SYSCALL(sbrk)
SYSCALL(sleep)
SYSCALL(uptime)
SYSCALL(wolfie) // <- Lab-3
```

4. 在`user.h`中添加用户程序调用函数

```
struct stat;
struct rtcdate;

// system calls
int fork(void);
int exit(void) __attribute__((noreturn));
int wait(void);
int pipe(int*);
int write(int, const void*, int);
int read(int, void*, int);
int close(int);
int kill(int);
int exec(char*, char**);
int open(const char*, int);
int mknod(const char*, short, short);
int unlink(const char*);
int fstat(int fd, struct stat*);
int link(const char*, const char*);
int mkdir(const char*);
int chdir(const char*);
int dup(int);
int getpid(void);
char* sbrk(int);
int sleep(int);
int uptime(void);
int wolfie(void*, uint); // <- Lab-3

// ulib.c
int stat(const char*, struct stat*);
char* strcpy(char*, const char*);
void *memmove(void*, const void*, int);
char* strchr(const char*, char c);
int strcmp(const char*, const char*);
void printf(int, const char*, ...);
char* gets(char*, int max);
uint strlen(const char*);
void* memset(void*, int, uint);
void* malloc(uint);
void free(void*);
int atoi(const char*);
```

5. 在`sysproc.c`中编写`sys_wolf`函数

```
int
sys_wolfie(void)
{

  char img[] = "\
LGLfjjjtttjt;;;;;;;;,,,,,,,;,;;;;i;;;tftttjjLGGLff\n\
LLGftiiiiii,,;;;,;;,,,,,,,,,,,,,;;;;;,;fiitjLGLfff\n\
fLLftiiii;,,,;,,;,,,,,,,,,,,,,,,;;,;,,,fitjLLLLff\n\
LLLLjtiti,,,;,,,,,;,,,,,,,,,,,,,,;,,,;,,,jjfLLLLLL\n\
LLGGLfff,;,;;,,,,,;,,,,,,;,,,,,,,,,;,,;,,,LLGLLLGG\n\
GDDDGGG;;,,,,,,,,;,,,,,,,;,,,,,,,,,,;,,;,,;LDDGGDD\n\
KKKKEE;,,,;,,,,,,;,,,,,,,,,,,,,,,,,,;,,,,,,;EKKKKK\n\
KKKKEt,,,,;,,,,,,i,:::,,,:,,,,,,,:,:,,,,,,,,fKKKKK\n\
DDDDG,,,,;;,,,::,i,:::::::;:::::::,::,,,,,,,,DDDGD\n\
LLDKi,,,,;,,::::,i,:::::::;::::::::,::,,,;,,,iLLLL\n\
fDWG,,,,,,,,,:::,i,:::::::,:::,,,,:,::,,,,,,,,Lfff\n\
fLWt,,,,;,,,,:::,t,:::::,,,,:,,,,,,,,,,,,,;,,,ifff\n\
ffE;,,,,;,,,,,::,t,,,,,,,,,,:,,,,,,:,,,,,,,;,,;Lff\n\
fff,,,,,;,,,,,:,,j,,,,,,,,,;:,,;,,,,,,,,,,,;,,,Lff\n\
LEt,,,,,;,,,,,,,,f,,,,,,,,,;,,,,,,,,,,,,,,,,,,,iff\n\
WKi;,,,;;,,,,,,,,ji,,,,,,,,,,,,,;,,,,,,,,,,,,,,;Lf\n\
KDi,,,,,;,,,,,,:t;j,,,,,,,;,i,,,i,,,,,,,,,,;,,,;Lf\n\
Kfi;,,,;;,,,,,,,j:;,,,:,:,,,,i,,t,,,,,,,,,,;,,,,fL\n\
Wii,,,,;;,,,,,,t:::j,,::::,;,j,,jt,;,,,,,,,;,,,;tG\n\
K;i,,,,;i,;,,,i;::.:i::,:,,,i,;,ji,,,,,,,,,i;;,;iE\n\
K;i,,,;;ii,,,it:...:t,,:::,,t,j,L:j,i,,,,,,i;;;;;E\n\
Dii;,,;,f;,,it,::::.:j:,,::t,f,i,::i,j,,,;,i,;;;;D\n\
L;i;;,;,j,,jt,,,ii;,::t,:::i,;tj::;ttjGi;t;ti;;,iG\n\
f;;;;,,,jtj;:::.:.::;;,i::,,i,jjf,::,;ij;,tit;;;if\n\
ti;;,;;,jt,:::,;;,,i;::;t,:,t,;jtfji;,jttfjit;,;tj\n\
jiii,;;,i,:,jEWWKWKKf.::;t,,:;j;GGKKKGjjiif;;;;itj\n\
jtj;;;,,,LGEEEDGjjttji.:.,j,:tiLLLDEEKWWEjii;;;ttt\n\
jjj;;;;;;f:::::jiiiit:::.:tf,,;:::jttttLfiij,;;tjt\n\
tjtti;;;;t:::::iiii;i..:.:ttLii:::jiiiit::ft;;ttjt\n\
tfjtt;;;;;i::::,i;;;i:..::,i::i,::t;iiii::L;;tttft\n\
tLtttt;;;tj:::::i,,;;:...::...:..:,;;;;,::j;itttGj\n\
jGttttti;;ti:::::;i,:....:.....::::i;;;::iitttttDL\n\
GDfttttti;if;::::::::::...::....:::::::::jttttttDG\n\
DDGtttttttttif:::::::.....;.....::::::::,ftttttjDD\n\
GGGjtttttttttf:::::.......:.......::::::;tttjttfGG\n\
LLffjtttjjttttj::.:...............::::::ftttttjLGG\n\
LG;ftjtttjLjttij::...:............:::::,tttjtttLLL\n\
Lf,ffjjttttjLLjt;::.................:::fttfttfjLLL\n\
Li;jLjjftttttf::...................:::;ttfttjtLfff\n\
f,ijjLtfftttttj::.::.....::.......::::ttftttffffff\n\
;;jjjjLtLLititt::..:......::...:..:::,iitttGttffff\n\
,;fjjjjLjLLtiiii:::.............:::::tfjttffG,fjjj\n\
,tjjjjjjffj.fttt:::..:...........:::j:LtttGfj;jjjj\n\
,jjjjjjjjt.jfit;::::::.........::::tjLEjtLffj;tttt\n\
ijjjjjjjjtt ttftjt,:::....:...:::ijjjjjtjfjjj;iiii\n\
jfjjjjjjjtt itjjjj;i,:.::..::::ijjjjjjftLfjjj;;;;;\n\
jjjjjjjjjjt ,jjjjj;,;i;:::::,i;jjjjjjjftffjjj;;;;;\n\
\n\
";

  void *buf;
  uint size;
  // Fetch the arg pointer and content
  if((argptr(0, (void*)&buf, sizeof(*buf))) < 0 || (argint(1, (int*)&size)) < 0) {
    return -1;
  }

  if(size < sizeof(img)) {
    return -1;
  }

  strncpy((char*)buf, img, sizeof(img));

  return sizeof(img);
}
```

6. 添加用户程序`wolfietest.c`

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

int
main(void)
{
    char buf[3000];

    printf(1, "sys_call wolfie, return: %d\n", wolfie((void*)buf, 3000));
    printf(1,"%s", buf);

    exit();
}
```

7. 在`Makefile`中添加对`wolfietest.c`的引用

```
UPROGS=\
    _cat\
    _echo\
    _forktest\
    _grep\
    _init\
    _kill\
    _ln\
    _ls\
    _mkdir\
    _rm\
    _sh\
    _stressfs\
    _usertests\
    _wc\
    _zombie\
    _wolfietest\ # <- Lab-3

...

EXTRA=\
    mkfs.c ulib.c user.h cat.c echo.c forktest.c grep.c kill.c\
    ln.c ls.c mkdir.c rm.c stressfs.c usertests.c wc.c zombie.c wolfietest.c\
    printf.c umalloc.c\
    README dot-bochsrc *.pl toc.* runoff runoff1 runoff.list\
    .gdbinit.tmpl gdbutil\
```

实验结果

因为是在docker中进行的实验，所以qemu没有图形化界面，使用`make qemu-nox`来测试

输出结果

```
SeaBIOS (version 1.13.0-1ubuntu1.1)


iPXE (http://ipxe.org) 00:03.0 CA00     PCI2.10 PnP PMM+1FF8CA10+1FECCA10 CA00



Booting from Hard Disk..xv6...
cpu1: starting 1
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200   nlog 30 logstart 2 inodestart 32 bmap
start 58
init: starting sh
$ ls
.              1 1 512
..             1 1 512
README         2 2 2286
cat            2 3 16268
echo           2 4 15120
forktest       2 5 9436
grep           2 6 18484
init           2 7 15704
kill           2 8 15148
ln             2 9 15004
ls             2 10 17636
mkdir          2 11 15248
rm             2 12 15228
sh             2 13 27860
stressfs       2 14 16136
usertests      2 15 67244
wc             2 16 17004
zombie         2 17 14816
wolfietest     2 18 14972
```

```
console        3 19 0
$ wolfietest
sys_call wolfie, return: 2399
LGLfjjjtttjt;;;;;;;;,,,,,,,;,;;;;i;;;tftttjjLGGLff
LLGftiiiiii,,;;;,;;,,,,,,,,,,,,,;;;;;,;fiitjLGLfff
fLLftiiii;,,,;,,;,,,,,,,,,,,,,,,,;;,;,,,fitjLLLLff
LLLLjtiti,,,;,,,,,;,,,,,,,,,,,,,,,;,,;,,,jjfLLLLLL
LLGGLfff,;,;;,,,,,;,,,,,,;,,,,,,,,,;,,;,,,LLGLLLGG
GDDDGGG;;,,,,,,,,;,,,,,,,;,,,,,,,,,;,,;,,,;LDDGGDD
KKKKEE;,,,;,,,,,,;,,,,,,,,,,,,,,,,,,;,,,,,,;EKKKKK
KKKKEt,,,,;,,,,,,i,:::,,,:,,,,,,,,:,:,,,,,,,fKKKKK
DDDDG,,,,;;,,,::,i,:::::::;:::::::,::,,,,,,,,DDDGD
LLDKi,,,,;,,::::,i,:::::::;:::::::,::,,,,;,,,iLLLL
fDWG,,,,,,,,,:::,i,:::::::,:::,,,,:,::,,,,,,,,Lfff
fLWt,,,,;,,,,:::,t,::::,,,,,:,,,,,,,,,,,,,;,,,ifff
ffE;,,,,;,,,,,::,t,,,,,,,,,,:,,,,,,:,,,,,,;,,,;Lff
fff,,,,,;,,,,,:,,j,,,,,,,,,;:,,;,,,,,,,,,,,;,,,Lff
LEt,,,,,;,,,,,,,,f,,,,,,,,,;,,,,,,,,,,,,,,,,,,,iff
WKi;,,,;;,,,,,,,,ji,,,,,,,,,,,,,;,,,,,,,,,,,,,,;Lf
KDi,,,,,;,,,,,,:t;j,,,,,,,;,i,,,i,,,,,,,,,,;,,,;Lf
Kfi;,,,;;,,,,,,,j:;,,,:,:,,,,i,,t,,,,,,,,,,;,,,,fL
Wii,,,,;;,,,,,,t:::j,,::::,;,j,,jt,;,,,,,,,;,,,;tG
K;i,,,,;i,;,,,i;::.:i::,:,,,i,;,ji,,,,,,,,,i;;,;iE
K;i,,,;;ii,,,it:...:t,,:::,,t,j,L:j,i,,,,,,i;;;;;E
Dii;,,;,f;,,it,::::.:j:,,::t,f,i,::i,j,,,;,i,;;;;D
L;i;;,;,j,,jt,,,ii;,::t,:::i,;tj::;ttjGi;t;ti;;,iG
f;;;;,,,jtj;:::.:.::;;,i::,,i,jjf,::,;ij;,tit;;;if
ti;;,;;,jt,:::,;;,,i;::;t,:,t,;jtfji;,jttfjit;,;tj
jiii,;;,i,:,jEWWKWKKf.::;t,,:;j;GGKKKGjjiif;;;;itj
jtj;;;,,,LGEEEDGjjttji.:.,j,:tiLLLDEEKWWEjii;;;ttt
jjj;;;;;;f:::::jiiiit:::.:tf,,;:::jttttLfiij,;;tjt
tjtti;;;;t:::::iiii;i..:.:ttLii:::jiiiit::ft;;ttjt
tfjtt;;;;;i::::,i;;;i:..::,i::i,::t;iiii::L;;tttft
tLtttt;;;tj:::::i,,;;:...::...:..:,;;;;,::j;itttGj
jGttttti;;ti:::::;i,:....:.....::::i;;;::iitttttDL
GDfttttti;if;:::::::::::...:...::::::::::jttttttDG
DDGttttttttif::::::::.....;.....::::::::,ftttttjDD
GGGjtttttttttf::::........:.......::::::;tttjttfGG
LLffjtttjjttttj::.:...............::::::ftttttjLGG
LG;ftjtttjLjttij::...:............:::::,tttjtttLLL
Lf,ffjjttttjLLjt;::.................:::fttfttfjLLL
Li;jLjjftttttf::...................:::;ttfttjtLfff
f,ijjLtffttttj::.::......::........:::ttftttffffff
;;jjjjLtLLititt::..:......::...:..:::,iitttGttffff
,;fjjjjLjLLtiiii::..............:::::tfjttffG,fjjj
,tjjjjjjffj.fttt::..:............:::j:LtttGfj;jjjj
,jjjjjjjjt.jfit;::::::.........::::tjLEjtLffj;tttt
ijjjjjjjjtt ttftjt,:::....:...:::ijjjjjtjfjjj;iiii
jfjjjjjjjtt itjjjj;i,:.::..::::ijjjjjjftLfjjj;;;;;
jjjjjjjjjjt ,jjjjj;,;i;:::::,i;jjjjjjjftffjjj;;;;;
```

```
$
```