Sistema-Deiak: Linux Kernel API

Kepa Bengoetxea
kepa.bengoetxea@ehu.es

Erreferentziak

- API: Application Programming Interface
  - en.wikipedia.org/wiki/Application_programming_interface
- POSIX : Portable Operating System Interface
  - en.wikipedia.org/wiki/POSIX
- C POSIX library:
  - en.wikipedia.org/wiki/C_POSIX_library
- Syscalls:
  - en.wikipedia.org/wiki/System_call

Sistema eragilea

- Ez dago Sistema Eragilearen(SE) definizio borobilik
- Objektiboki, SEa *hardware*tik(HW) gertuen dagoen *software*a da. Helburuak:

1- HWaren konplexutasunaz abstrakzioa egin⇒ Interfaze edo alegiazko makina, HW hutsa baino erabilgarriagoa

2- Sistema informatikoaren funtzionamendu zilegia ziurtatu ⇒ Baliabide guztien kudeaketa orekatua (PUZa, Memoria, S/I disp.)

Sistema eragilea

- Zer da SEa? Programadorearen ikuspuntu funtzionala: makinaren zehaztasunak alde batera utziz haren baliabideak erabiltzeko aukera eskaintzen duen errutina multzoa ⇒ alegiazko makina
- SEaren interfazearen osagaiak:

  - Sistema-deiak

  - Komando-interpretatzailea: testuzkoak, grafikoak...

Motibazioa

- SEek eskaintzen dituzten funtzioak aztertuko ditugu.
- **Sistema-deien interfazea** sistema eragilearen funtsezkotzat joko dugu hemendik aurrera, eta bera izango da alegiazko makinaren funtzioak definitzen dituena. Beraz, sistema-deien multzoak definitzen du sistema eragilearen oinarrizko interfazea, eta berau zehazten du iturburu-lengoaiaren mailan makinen arteko bateragarritasuna.

Motibazioa

- Erabiltzailearentzat, sistema informatikoa aplikazioak egikaritzeko plataforma bat da.
- Programak garatzeko orduan HWaren kontrol zehatza eraman beharko balu programatzaileak programatzea ikaratzeko moduko lana izango litzateke.

Motibazioa

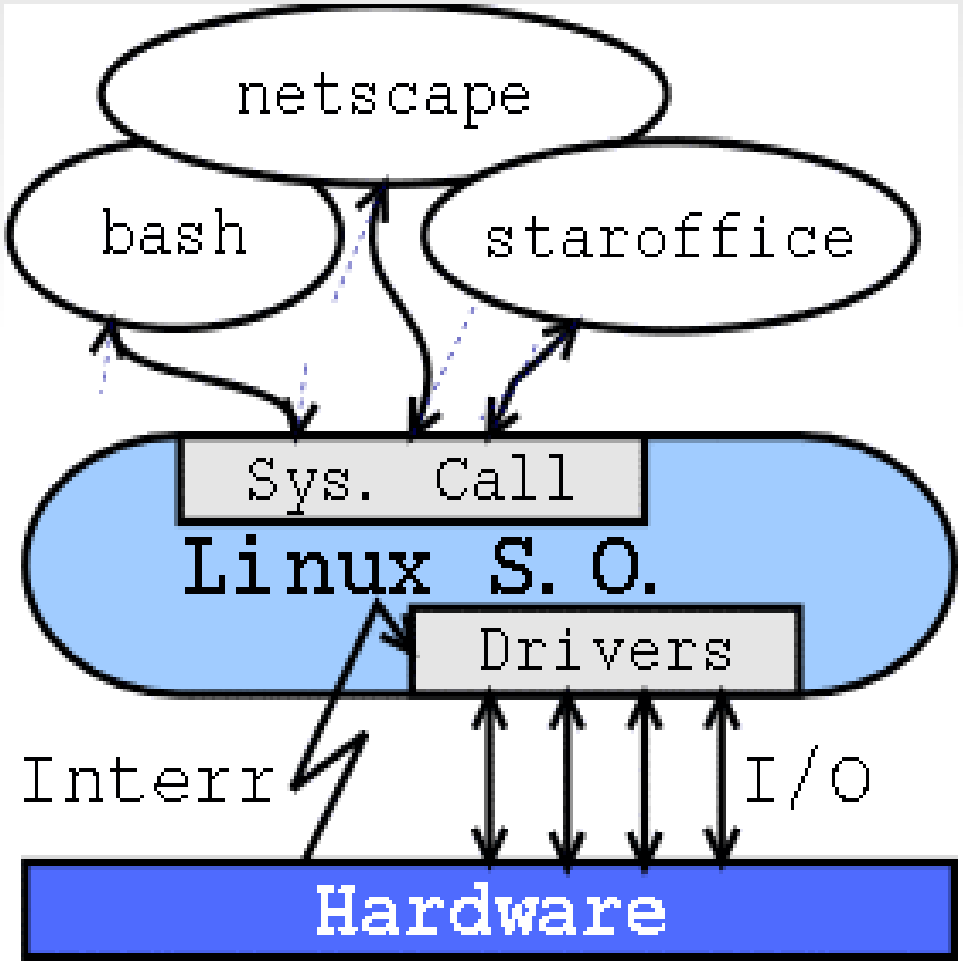
- Programazioa errazteko SEak sistema-deiak eta berarekin batera utilitate-programa multzo zabal bat eskaintzen du.
- Utilitate-programek programatzaileari laguntzeko zerbitzu anitz eskaintzen dituzte: testu-editoreak, konpiladoreak, araztaileak (debuggers), liburutegiak, etab. Orokorrean ez dira sistema eragilearen partea, beraren gainean eraikita baitaude berarekin banatu arren.

Sistema Deiak: Linux Kernel API

- Sistema Deia (ingeleraz system call) aplikazio batek Sistema Eragileari zerbitzu bat eskatzeko erabiltzen da.
- Sistema Eragileak sistema-deien multzo bat eskaintzen du (linux-eko atentzio edo egoiliar errutina multzoa, linux Kernel API edo POSIX* API liburutegi bezala ere ezagutzen dena). Programazioa eta ordenagailuaren erabilera errazten dutenak. Hardware baliabideak: PUZ, Memoria, Diskoa eta abarrak kudeatuz, eta horri esker, ordenagailuari etekin handiagoa ateraz.

*POSIX:Portable Operating System Interface; X UNIX-etik dator.

Sistema Deiak: Linux Kernel API



Sistema Deiak: Linux Kernel API

- GNU C Library, **glibc** bezala ezagutzen dena, GNU-ko C-ko liburutegi estandarra da. Linux banaketan **libc6** bezala ezagutzen da.
- Instalatzeko: sudo apt-get install libc6-dev
- Libc6 liburutegi honetan:
  - Linux-eko Sistema Deiak
  - C-ko liburutegi estandarra dago
- Linux 3.x ia 300 sistema dei baino gehiago daude. Erabilienak: open, read , write, close, wait, exec, fork, exit eta kill. “unistd.h” aurkitu dezakegu euren erazagupena.

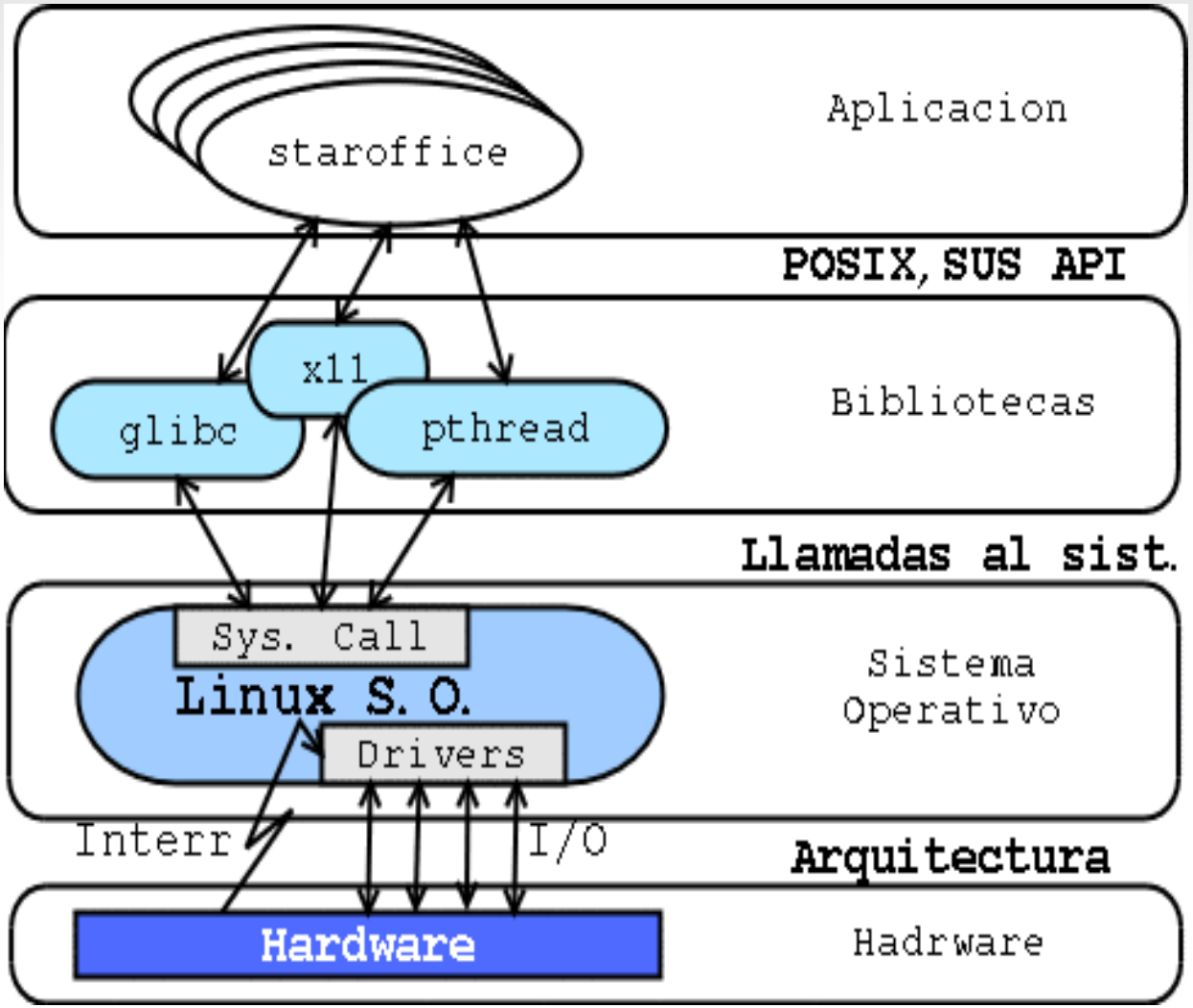
Sistema Deiak: Linux Kernel API

Orain arte 1 eta 3 atalak landu ditugu, orain 2 atala landuko dugu.

Man atalak

Atala	Deskribapena
1	Exekutatzeko programak eta shell-en aginduak
2	Sistema-deiak(kernelak zerbitzatutakoak)
3	C Liburutegiko oinarrizko funtzio deiak
4	Artxibo bereziak
5	Artxibo formatuak eta konbentzioak
6	Jokoak
7	Makro paketeak eta konbentzioak
8	Rootengandik exekutatzeko aginduak
9	Nukleo funtzioak

Sistema Deiak: Linux Kernel API



Sistema Deiak: Linux Kernel API

- Ze zerbitzu eskaintzen ditu kernelak?
  - Prozesuen kudeaketarako: fork, exec, setuid, getuid...
  - Fitxategien kudeaketarako: read, write, open, close, mkdir...
  - Memoriaren kudeaketarako: mmap, sbrk, munmap, mlock.
  - Sare zerbitzuen kudeaketarako: sethostname, setdomainname...
  - Seinaleen kudeaketarako: sigaction, sigsuspend,...

Sistema Deiak: Linux Kernel API

Kodea emanda: C liburutegiak erabiliz

```
#include <stdio.h>

int main(){int i;

for (i=0;i<=5;i++)

{printf("%d",i);}

return 0;}
```

Nola izango litzateke sistema-deiak edo linuxen APIak(Application Programming Interface) zuzenean erabiliz?

Sistema Deiak: Linux Kernel API

“write” bezalako sistema-deiak nola funtzionatzen duten ikusteko man erabili daiteke:

man 2 write (non 2 sistema-deien atala da)

SINOPSIA: #include <unistd.h>

ssize_t write(int fd, const void *buf, size_t num);

DESKRIBAPENA: write-k, buf-en hasitako buferretik, fd fitxategiko deskribatzaileak adierazitako fitxategian num bytes-etaraino idazten du.

ITZULITAKO BALOREA: (>0 bada, idatzitako bytes kopurua; 0 bada, ez dela ezer idatzi; -1 bada errorea eta errno aldagaian erroreari dagokion balioa jarriko da).

Sistema Deiak: Linux Kernel API

man 3 sprintf

SINOPSIA: #include <stdio.h>

int sprintf(char *str, const char *format, ...);

DESKRIBAPENA: str karaktere kate batean, parametrotzat pasatzen zaiona formatuarekin idazten du.

ITZULITAKO BALOREA: Funtzioak bihurtutako karaktere zenbakia itzultzen du

Sistema Deiak: Linux Kernel API

Kodea emanda: Kernelen sistema-deiak erabiliz

```
#include <unistd.h>
#include <stdio.h>
#include <string.h>
int main() {int i;char cad[10];
for (i=0;i<=5;i++)
        {sprintf(cad,"%d",i);
        write(1,cad,strlen(cad));
        }
return 0;}
```

Sistema Deiak: Linux Kernel API

```
#include <stdio.h>

int main(void)

{ printf("hello"); return 0; }
```

```
$gcc -o hello hello.c -static

$strace ./hello
```

strace - trace system calls and signals

```
#execve()  ejecuta  el programa indicado por filename
execve("./hello", ["./hello"], [/* 29 vars */]) = 0

...
write(1, "hello", 5hello)                    = 5
exit_group(0)                          = ?
```

Sistema Deiak: Erroreak kudeatu

Sistema deiek huts egiten dutenean, -1 balioa itzultzen dute, eta akatsaren zenbakia errno aldagaian gordetzen dute. perror() funtzioaren bitartez errno zenbakiari dagokion errore mezua pantailaratu dezakegu. man 3 perror

```
#include <stdio.h>

void perror(const char *s);
```

DESCRIBAPENA: perror() errutinak, errore estandar irteerara doan mezu bat sortzen du, sistema-dei batean zehar edo zenbait liburutegi funtziotan aurkitutako azken errorea deskribatuz.

Sistema Deiak: Erroreak kudeatu

```
/* erroreak.c * 53 lehen erroreak zerrendatzen ditu:
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
int main(){int i;
for(i=0;i<=53;i++)
{fprintf(stderr,"%3d",i);errno=i; perror(" ");}
exit(0);}
```

Sistema Deiak: Erroreak kudeatu

0 : Success

1 : Operation not permitted

2 : No such file or directory

3 : No such process

4 : Interrupted system call

5 : Input/output error

6 : No such device or address

7 : Argument list too long

8 : Exec format error

....

Sistema Deiak: Erroreak kudeatu

```
#include <sys/types.h>
#include <sys/stat.h>
#include <stdlib.h>
#include <fcntl.h>
#include <stdio.h>
int main(int argc, char *argv[])
{
int fd;fd=open(argv[1],O_RDWR);
if (fd==-1)
{perror("open ");
exit(-1);
}
printf( "Irekitako fitxategiak,
    honako deskribatzailea du
    %d.\n", fd);
close( fd );
return 0;}
```

Sistema Deiak: Fitxategiak kudeatzeko

fd = open (file, modo apertura r|w...[, permisos]) – idazteko/irakurtzeko fitxategia zabaldu

s = close (fd) – fitxategi bat itxi

n = read (fd, buffer, nbytes) – fitxategitik irakurritako nbyte buffer-ean gorde.

n = write (fd, buffer, nbytes) – bufferreko nbyte idatzi fitxategi batean.

pos = lseek (fd, offset, whence) – fitxategiaren erakuslea mugitu.

s = stat (name, &buf) – fitxategi baten izenarekin, i-nodo informazioa lortzeko

n= chmod(file, permisos)- fitxategi baten baimenak aldatzeko

...

Sistema Deiak: Fitxategiak kudeatzeko

- Norbaitek C liburutegiko oinarrizko funtzioak gustuko ez baditu, linuxen sistema-deiak erabil ditzake. Adb:
  - fwrite erabili beharrean
  - Zure fwrite sortu dezakezu ,write sistema-deia erabiliz

Sistema deiak: Memoria babestua

- Sistema deia SEari zerbitzu bat eskatzeko erabiltzen da. Sistema deiaren kodea exekutatzeko software eten bat gertatzen da.
- x86 konputagailuetan eten sistema mota bi daude:
  - Hardware eteneak
  - Software etenak edo ezepzioak

Sistema deiak: Memoria babestua

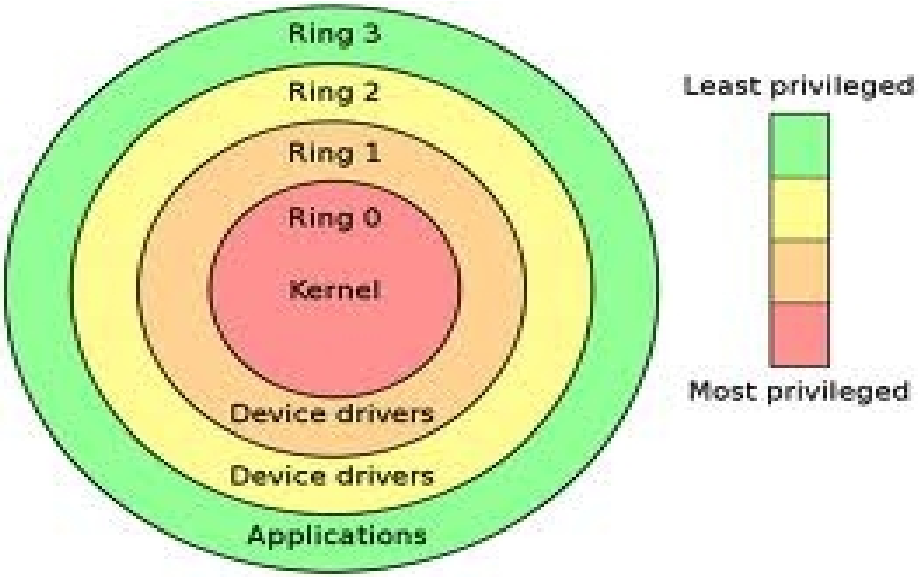
- Hardware etenak: Gailuak eta PUZa edo PUZak euren artean komunikatzeko erabiltzen dira. Ez daukate zer ikusirik momentuan exekutatzen dagoan prozesuarekin. Hau da, gailu batek zeozer behar duela adierazteko PUZari IRQ (“Interrupt Request”) bat bidaltzen dio, eta horrela PUZak egiten ari dena usten du, gailuaren eskaera betearazteko. Hardware etenen mekanismoak PUZa S/Iko gailuen egoerataz etengabe galdezka ibiltea ekiditzen du (Gailuak zeozer nahi izanez gero eten bat bidaltzen baitiote PUZari)

Sistema deiak: Memoria babestua

- Software etenak
  - sistemaren baliabideak erabiltzeko kernelak eskeintzen dituen zerbitzuei edo sistema deiei deitzeko erabiltzen dira.
  - Baita gure prozesuak sortutako ezepzioei erantzuteko, hau da, divide-by-zero, Arithmetic Logic Unit sortua edo illegal address, Memory Management Unit sortua.

Sistema deiak: Memoria babestua

- x86 ordenagailuak kodea exekutatzeko 4 modu ezberdin daukate: zero, bat, bi eta hiru eraztunetan. Linuxen "gure kodea" hirugarren eraztunean exekutatzen da eta Hardware edo Software eten bat sortzen denean "kernel kodea" zero eraztunean exekutatzen da. Sistema deiari deitzerakoan hirutik zerora salto egiten dugu. Zero eraztunean kodea edozein helbide atzitu dezake. Hiru eraztunean kodeak soilik gure aplikazioaren memoria gunera soilik atzitzeko baimena du.



Sistema deiak: Memoria babestua

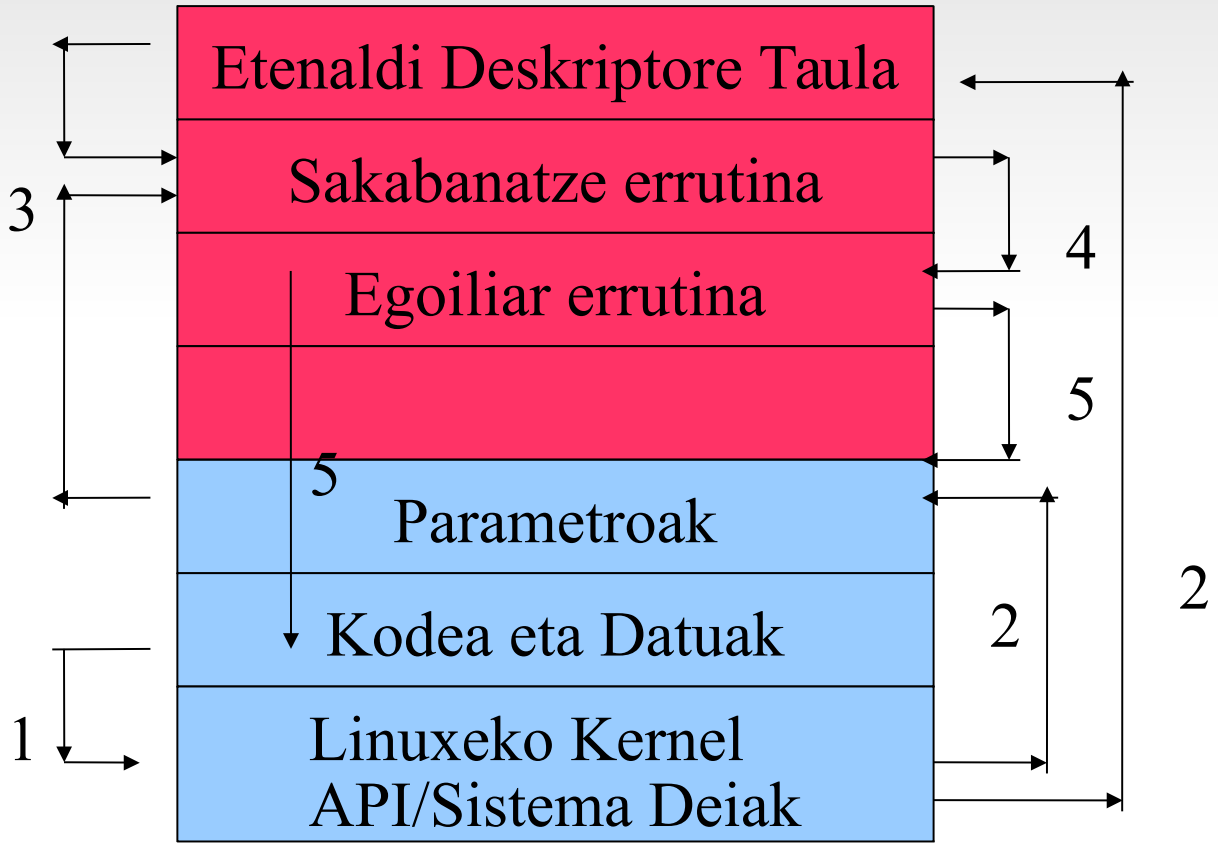
Linuxeko zerbitzu bati edo atentzio edo egoiliar errutinei deitzerakoan, linux-ek indirekzio teknika erabiltzen du. Teknika honek, Etenaldi Deskriptoreen Taula (EDT) bat erabiltzen du. EDT honetan ze etenaldiari ze atentzio errutina dagokion adierazten da. EDTaren sarrerak gutxitzeko, sarrera/irteerako errutinak, soft erroreko errutinak, eta abar... ,sakabanatze errutinatan taldekatzen dira. Horregatik, eten bat sortzen denean sakabanatze errutina bati deitzen zaio, parametrotzat ze atentzio errutinari deitu behar dion pasatuz.

Sistema deiak: Memoria babestua

Sakabanatze errutinatik dagokion atentzio errutinari deitzen zaio.

Atentzio egoiliar errutinaren exekuzioa bukatu ondoren, kontrola erabiltzailearen programari ematen zaio, emaitza parametroetan utziz.

Sistema-Deiak: Memoria babestua



Sistema deiak: Memoria babestua

- EDT, beheko memorian kokatzen da eta 1024 byte ditu, hori dela eta, 4 byteko 256 sarrera soilik eduki dezake, sarrera bakoitza errutina egoiliar edo sakabanatze errutina baten helbidea izango du.
- EDT honen lehenengo betebeharra BIOS zerbitzuen errutinen helbideak izatea da. Baina hauek ez dute sarrera asko betetzen. Beste gainerako sarrerak sistema zerbitzuekin betetzen dira.

Sistema deiak: Memoria babestua

- EDTn dauden BIOSeko sarrerak BIOSaren memoriara apuntatzen dute (helbide hauek berdinak dira edozein SEarentzako (linux, windows, mac...) eta gainerako sarrerak memoria RAMean dauden SEaren errutinei apuntatzen diete.
- Sistema deiak 80. etena erabiltzen du, sakabanatze errutinari deituz parametro baten bidez pasatuz, zein errutina egoiliar exekutatu nahi duen. Errutina hauek “modu kernelean” edo “zero eraztunean” exekutatzen dira.

Sistema deiak: Memoria babestua

- Sistema deien bitartez bakarrik, sistema eragileen errutinak erabili ditzakegu. Sistema honekin memoria, erabiltzaile programetatik babesten dugu.
- Erabiltzaile programa batek, sistema eragilearen errutina baten helbidean sartuz gero eta bertan kodea moldatuz gero, beste programa guztiak ez ziran ondo ibiliko = birusa. Hau dela eta, erabiltzaile programa bat ezin da sistema eragilearen memorian zuzenean sartu. Zuzenean sar daitekeen leku bakarra bere datu eta instrukzio gunera da.

Sistema deiak: Memoria babestua

- Sistema dei bat burutzerakoan: erabiltzaile modutik, gainbegirale modura, aldatzen da. SEaren errutina exekutatu ondoren, erabiltzaile modura bueltatuko da.
- Erabiltzaile moduan dagoenean, hardware mekanismoaren bidez kontrolatzen da memoriaren sarbidea. Gainbegirale moduan, berriz, hardware mekanismoa desaktibatzen da.

Sistema deiak: Memoria babestua

- Prozedura hau, jatetxe baten jantoki batena gogorarazten dit. Jantokian, bezero guztiek zerbitzariari nahi dutena eskatzen diote, baina ez dira inoiz sukaldean sartzen. Zerbitzariak, sukaldetik pasa ondoren, bezeroak eskatutako platerra ekarriko du. Jankideak ezingo dute sukaldea hondatu, ezin direlako honetan sartu.

Sistema deiak: Memoria babestua Laburbilduz

- Erabiltzaile kodea:
  - Ezin da ordenagailuaren baliabideetara zuzenean sartu
- Kernel kodea :
  - Sistema osorako sarbidea, ordenagailuaren baliabide guztiak kudeatuz
  - Edukia: gailu kontrolatzaileak, memoria gestioa, fitxategi sistema,...

Sistema deien mekanismoa ikusgai

Ze zerbitzu eskaintzen ditu kernelak?

$uname -r ->3.13.0-46-generic bertsioan, /usr/include/asm-generic/unistd.h artxiboan zenbakitutako sistema deiak existitzen dira.

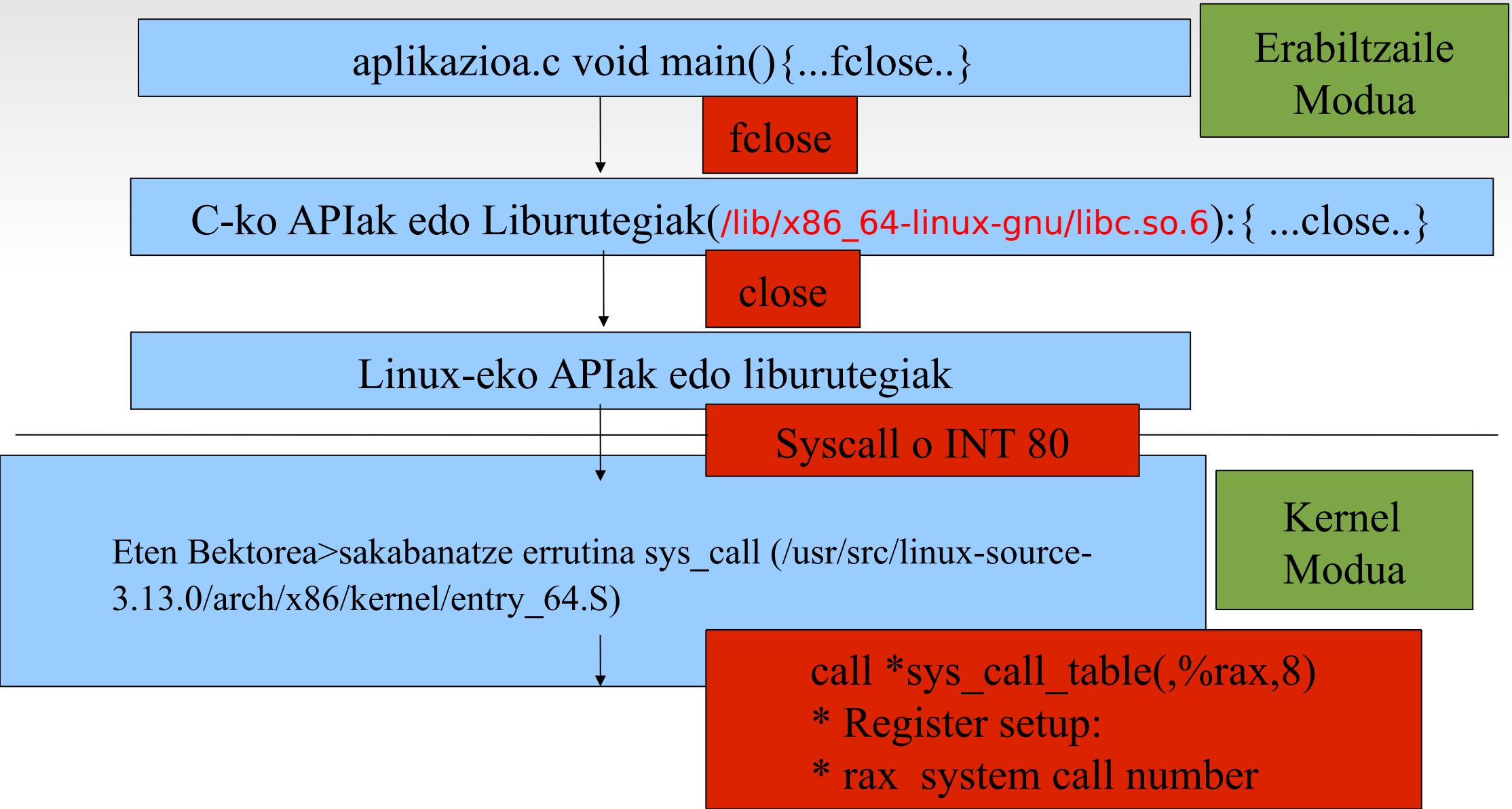
Fitxategi honetan esaten da, funtzio honen kodea ze fitxategian aurki genezake:

```
/* fs/read_write.c */

#define __NR3264_lseek 62

__SC_3264(__NR3264_lseek, sys_llseek, sys_lseek)

#define __NR_read 63

__SYSCALL(__NR_read, sys_read)

#define __NR_write 64

__SYSCALL(__NR_write, sys_write)
```

Sistema deien mekanismoa ikusgai

- Linux-eko kodea ikusi ahal dugu? Bai. Nola?
- sudo apt-get install linux-source
- aurreko komandoarekin /usr/src katalogora linux-en iturriak jaisten ditugu, hau da, linux-source-3.13.0.tar.bz2 fitxategia.
- Mugitu hurrengo katalogora:$ cd /usr/src
- Hurrengo komandoarekin, urrats batean, fitxategia deskonprimitu eta paketea zabalduko dugu:
- $ sudo tar jxvf linux-source-3.13.0.tar.bz2

Sistema deien mekanismoa ikusgai



Sistema deien mekanismoa ikusgai

system_call_fastpath:
call *sys_call_table(,%rax,8)

(rax=57->sys_close) beren kodea

/usr/src/linux-source-3.13.0/fs/open.c dago hemen /usr/include/asm-generic/unistd.h esaten den moduan.

Kernel Modua

Fitxategi Sistema ->Hardware

Sistema deien mekanismoa ikusgai

Kode hau irakurri nahi baduzu, /usr/src/linux-source-3.13.0/arch/x86/kernel/entry_64.S fitxategian dago.

```
system_call_fastpath:
        cmpq $__NR_syscall_max,%rax
        ja badsys
        movq %r10,%rcx
        call *sys_call_table(,%rax,8)  # XXX:    rip relative
        movq %rax,RAX-ARGOFFSET(%rsp)
```

Sistema deien mekanismoa ikusgai

- Programa guztiek erabiltzeko, linuxeko sistema deiak (APIak) objektu modulu bezala liburutegian sartzen dira, errutinen izen eta erreferentziekin (unistd.h). Kode hau, arduratuko da parametroak PUZeko erregistroen bidez egoiliar errutinara pasatzen, eta 80 etenaren bidez instrukzioa edo sistema deia exekutatuz.