

# Hibakeresés és elhárítás Linux rendszeren

Mátó Péter <[atya@fsf.hu](mailto:atya@fsf.hu)>

# A hibák alaptípusai

## ■ Előre várt hibák

- A program képes kezelni a hibát
- A program nem képes kezelni

## ■ Váratlan hibák

- A programozók nem számítottak rá
- Ha a program megpróbálja orvosolni a hibát, az általában rosszabb, mintha a felhasználóra bízná a javítást
- Helyesebb, ha a program fail-safe módon működik

# A hibák megjelenése

Fontos megjegyezni: a programok általában valamilyen módon közlik a hibát, csak néha ez nem egyértelmű.

A hibajelzések típusai:

- Érthető hibaüzenet valamilyen csatornán
- Urdu nyelvű, de érthető hibaüzenet
- Egyértelmű, a hibára utaló bináris vagy számkód
- A hibát jellemző rendellenes működéssel „jelzi”
- A hibaüzenet nem érthető:

ed: -e kifejezés #1, karakter 43: Hibás hivatkozás (\1) a(z) `s' parancs RHS-ére

# A programok kommunikációs felülete

- Dokumentáció, levelezési listák
- parancssoros és grafikus: STDOUT, STDERR
- Grafikus, ncurses: tájékoztató ablakok, státuszsor, beépített napló
- Szerverek:
  - rendszernaplók
  - saját naplóállományok
- SMTP üzenetek (pl. cron, monitor rendszer)
- Egyéb jelzések (pl. hang)

# Az STDOUT és STDERR kimenetek

- Parancssoros programok felülete
- Grafikus programok is használják (terminálból indítva a legegyszerűbb megnézni)
- Hiba vagy sok kimenet esetén későbbi elemzésre érdemes elmenteni  
program 2>&1 >~/tmp/program.log
- A bőbeszédűség általában befolyásolható  
-v bőbeszédű (verbose) | -q csendes (quiet)

# A rendszernaplók

## Az /etc/syslog.conf

```
*.info;mail.none;authpriv.none;cron.none      /var/log/messages
authpriv.*                                       /var/log/secure
mail.*                                           -/var/log/maillog
cron.*                                           /var/log/cron
*.emerg                                          *
local7.*                                         /var/log/boot.log
```

# A naplóbejegyzések

```
Nov 17 23:31:20 fractal slapd[6862]: bdb_initialize:  
Sleepycat Software: Berkeley DB 4.1.25: (December 19,  
2002)  
Nov 17 23:31:20 fractal slapd[6862]: could not open  
config file "/etc/openldap/slapd.conf": No such file or  
directory (2)  
Nov 17 23:31:20 fractal slapd[6862]: slapd shutdown:  
freeing system resources.  
Nov 17 23:31:20 fractal slapd[6862]: slapd stopped.  
Nov 17 23:31:20 fractal slapd[6862]:  
connections_destroy: nothing to destroy.
```

# Naplózási szint

- A szint közvetlen beállítása (pl. sshd)

QUIET | FATAL | ERROR | INFO | VERBOSE | DEBUG

- A naplózás maszkolása (pl. OpenLDAP)

$$256 + 512 = 768$$

-1	enable all debugging
0	no debugging
1	trace function calls
2	debug packet handling
4	heavy trace debugging
8	connection management
...	
256	stats log connections/operations/results
512	stats log entries sent
...	

# Különös hibajelenségek

- A program elkezd nagyon sok memóriát fogyasztani
- A program érthetetlenül sok processzoridőt használ fel
- A program reprodukálhatóan egy bizonyos helyen leáll/elesik
- A program nem várt módon reagál
- Valamilyen sajátos, egyedi dolgot ír ki

# Az strace nyomkövető

## A rendszerhívások követését teszi lehetővé

- o fájl a kimenetet fájlba menti
- f a gyerekek követésére
- ff a gyereket fájl.pid formátumban
- F a vfork hívást is követi
- p pid a megadott processzt követi
- s n a sztringek megadott részét írja ki (def: 32)
- e filter szűrési feltétel (pl. trace=open v. -eopen v. trace=open,close,read,write v. trace=file network, ipc, signal...)

# Az ltrace nyomkövető

A könyvtári függvényhívások követését teszi lehetővé

- o fájl a kimenetet fájlba menti
- f a gyerekek követésére
- p pid a megadott processzt követi
- s n a sztringek megadott részét írja ki (def: 32)
- e filter szűrési feltétel (pl. printf)
- l lib a megadott könyvtár függvényeit írja ki

# Egy egyszerű, hibás program

```
#include<sys/stat.h>
#include<fcntl.h>

int main()
{
    int fd;

    mkdir("fa", 0755);
    chdir("fa");
    fd = open ("alma", O_WRONLY | O_CREAT | O_NONBLOCK |
                O_NOCTTY, S_IRUSR | S_IWUSR | S_IRGRP | S_IWGRP |
                S_IROTH | S_IWOTH);

    write(fd, "alma\n", 5);
    close(fd);
    return 0;
}
```

# Az első hibajelenség

Az ellenőrzés hiánya:

...

```
mkdir("fa", 0755)           = -1 EEXIST (File  
                             exists)
```

```
chdir("fa")                 = 0
```

...

# A második hibajelenség

## Jogosultsági probléma I.

```
...
mkdir("fa", 0755)    = -1 EEXIST (File exists)
chdir("fa")          = -1 EACCES (Permission denied)
open("alma", O_WRONLY|O_NONBLOCK|O_CREAT|
                  O_NOCTTY, 0666) = 3
write(3, "alma\n", 5) = 5
close(3)             = 0
...
```

# A harmadik hibajelenség

## Jogosultsági probléma II.

```
...
mkdir("fa", 0755)    = -1 EACCES (Permission denied)
chdir("fa")          = -1 ENOENT (No such file or
                      directory)
open("alma", O_WRONLY|O_NONBLOCK|O_CREAT|
      O_NOCTTY, 0666) = -1 EACCES (Permission denied)
close(-1)            = -1 EBADF (Bad file descriptor)
exit_group(0)        = ?
```

# A negyedik hibajelenség

## umask probléma

```
...
mkdir("fa", 0755)      = 0
chdir("fa")            = 0
open("alma", O_WRONLY|O_NONBLOCK|O_CREAT|
    O_NOCTTY, 0666) = -1 EACCES (Permission denied)
write(-1, "alma\n", 5) = -1 EBADF (Bad file
                                descriptor)
close(-1)              = -1 EBADF (Bad file descriptor)
exit_group(0)         = ?
```

# Idegen programok futtatása

```
$ ./knetfilter
```

```
./knetfilter: error while loading shared libraries: libkio.so.4:  
cannot open shared object file: No such file or directory
```

A probléma: nincs meg a rendszeren a megadott függvénykönyvtár

A probléma felderítése

- Az ldd használata
- A strace -eopen használata

# Idegen programok futtatása

## ■ ldd

libkio.so.4 => not found

libkdesu.so.4 => not found

libdl.so.2 => /lib/libdl.so.2 (0x00c4b000)

...

## ■ strace -eopen

open("/etc/ld.so.cache", O\_RDONLY) = 3

open("/lib/tls/i686/libkio.so.4", O\_RDONLY) = -1  
ENOENT (No such file or directory)

open("/lib/tls/libkio.so.4", O\_RDONLY) = -1 ENOENT  
(No such file or directory)

...

# Néhány hasznos apróság

A program által használt állományok megkeresése

```
$ touch /tmp/most  
$ program  
$ find / -newer /tmp/most
```

A program által használt állományok, vagy a használt állományok programjának keresése

```
$ lsof -p 1748  
$ lsof -i :137  
$ lsof /path/file
```

# Shell programok hibakeresése

- `sh -x` : minden végrehajtott parancsot kiír, így könnyebb követni, hogy hol lehet a hiba
- `sh -u` : a nem inicializált változókra figyelmeztet, ezzel sok hiba elkerülhető
- Egy gyakori gond: a program nem a várt környezetben fut (pl. cron)
  - Nincsenek meg a szükséges környezeti változók
  - Nem a megfelelő könyvtárban van
  - Nem a megfelelő felhasználó nevében fut

# Vas vagy driver problémák

## ■ Hasznos segédeszközök

- dmesg - a rendszermag üzeneteinek megtekintésére (néha csak a /var/log/boot.log segíthet)
- lspci - a PCI kártyák megnézésére

## ■ Memória hibáknál hosszas memtest futtatás

## ■ Diszk hibák a logból láthatók

- Ha kell, meg kell hajtani a diszket:  
`# dd if=/dev/hda of=/dev/null bs=1024k count=81920`
- fsck, mke2fs -c, badblocks

# boot gondok

- A régi kernelt mindig elérhetően kell hagyni
- Szükség esetén boot tetszőleges Linux CD-ről (Knoppix, Gentoo Live ...)
  - hasznos opció: `root=/dev/rootdev`
  - boot után fel kell csatolni a gép diszkjeit, és javítani a problémát (jól használható a `chroot` parancs)
- Lilo gondok
  - Általában a boot loader nem ott van, ahol a lilo futtatása idején (LI), lásd részletesen a Lilo kézikönyvben.

# Teljesítmény gondok

- Eszközök a problémák megoldására
  - Pocsessor:  
top, sar
  - Memória:  
free, sar -r, top, vmstat, cat /proc/meminfo
  - Diszk:  
sar -d, df -h, du -h
  - Hálózat:  
iptraf, tcpsray

# Egyéb hibák

Nem indul el vagy feleslegesen indul el egy szolgáltatás

- rc scriptek  
(Debian: update-rc.d, Gentoo: rc-update stb.)
- Nem megfelelő indítási sorrend
- Az indító scriptek kézi indítása
- Konfigurációs állományok ellenőrzése
- A korábban látott debug technikák használata

# Elfelejtettük a root jelszavát

- Valahogy hozzáférünk a rendszerhez:

- boot közben:  
linux init=/bin/bash
- Boot egy Live CD-ről

- Ezek után

- mount /dev/rootdev /mnt vagy  
mount -o remount,rw /
- vi /mnt/etc/shadow
- umount /mnt vagy  
mount -o remount,ro /
- reboot

Köszönöm a figyelmet

# Hibakeresés és elhárítás Linux rendszeren

Mátó Péter <[atya@fsf.hu](mailto:atya@fsf.hu)>