Chapter 03. 프로세스 모니터링

프로세스 모니터링 도구와 프로세스 분석

프로세스 모니터링

프로세스의 실행과 관리 및 디버깅

- 프로세스 실행하기
  - bg
  - fg
  - jobs
  - screen
- 프로세스 확인하기
  - ps
  - /proc/<pid>/
  - pstree
- 프로세스 종료하기
  - kill
  - killall
- 프로세스 디버깅
  - 파일/디렉토리 사용 프로세스
    - lsof
    - fuser
  - 시스템콜 트레이싱
    - strace
  - 라이브러리 트레이싱
    - ltrace

프로세스 (백그라운드) 실행하기

fg (foreground), bg (background), jobs

- 백그라운드로 실행하기
  - 실행 시 결정
    - tail -f /var/log/syslog : 앞(?) 에서 실행, Foreground
    - tail -f /var/log/syslog & : 뒤(?) 에서 실행, Background
  - 실행 후 결정
    - tail -f /var/log/message : 실행
    - tail -f /var/log/message <ctrl + z> : 실행 중, 중단 (ctrl+z)
    - jobs : 중단/백그라운드 실행중인 프로세스 확인
    - bg %1 : 중단 된 프로세스 백그라운드에서 실행
  - 관리 및 foreground 전환
    - jobs
    - fg %1 : 다시 foreground 로 전환
    - <ctrl + c> 로 종료 - 또는 kill %1 으로 강제 종료

```
Jun 13 12:28:26 user1-VirtualBox dbus-daemon[3003]: [session u
Successfully activated service 'org.gnome.Terminal'
Jun 13 12:28:26 user1-VirtualBox systemd[2978]: Started GNOME
Jun 13 12:28:30 user1-VirtualBox org.gnome.Shell.desktop[3229]
arning: Overwriting existing binding of keysym 73 with keysym
^Z
[1]+  정지됨                 tail -f /var/log/syslog
user1@user1-VirtualBox:~$ jobs
[1]+  정지됨                 tail -f /var/log/syslog
user1@user1-VirtualBox:~$ bg %1
[1]+ tail -f /var/log/syslog &
user1@user1-VirtualBox:~$ jobs
[1]+  실행중                 tail -f /var/log/syslog &
user1@user1-VirtualBox:~$ fg %1
tail -f /var/log/syslog
```

프로세스 확인 - ps (process status)

ps - 프로세스 관리를 위한 all-in-one 유틸리티

- 프로세스 상태 코드
  - D : Uninterruptible sleep (usually IO) - IO 대기상태
  - R : Running - 실행 중 상태
  - S : Interruptible Sleep (waiting for an event to complete) - 깨울 수 있는 대기 상태
  - T : Stopped, either by a job control signal or because it is being traced. - 중지 된 상태 (작업 제어 신호나 트레이싱 시그널로 인함)
  - Z : Defunct (“zombie”) process, terminated but not reaped by its parent - 좀비 프로세스 상태, 종료 되었으나 부모 프로세스에 의해 처리되지 않음.

  - < : high priority
  - N : log priority
  - L : pages locked into memory
  - s : session leader
  - l : multi-threaded
  - + : foreground process group

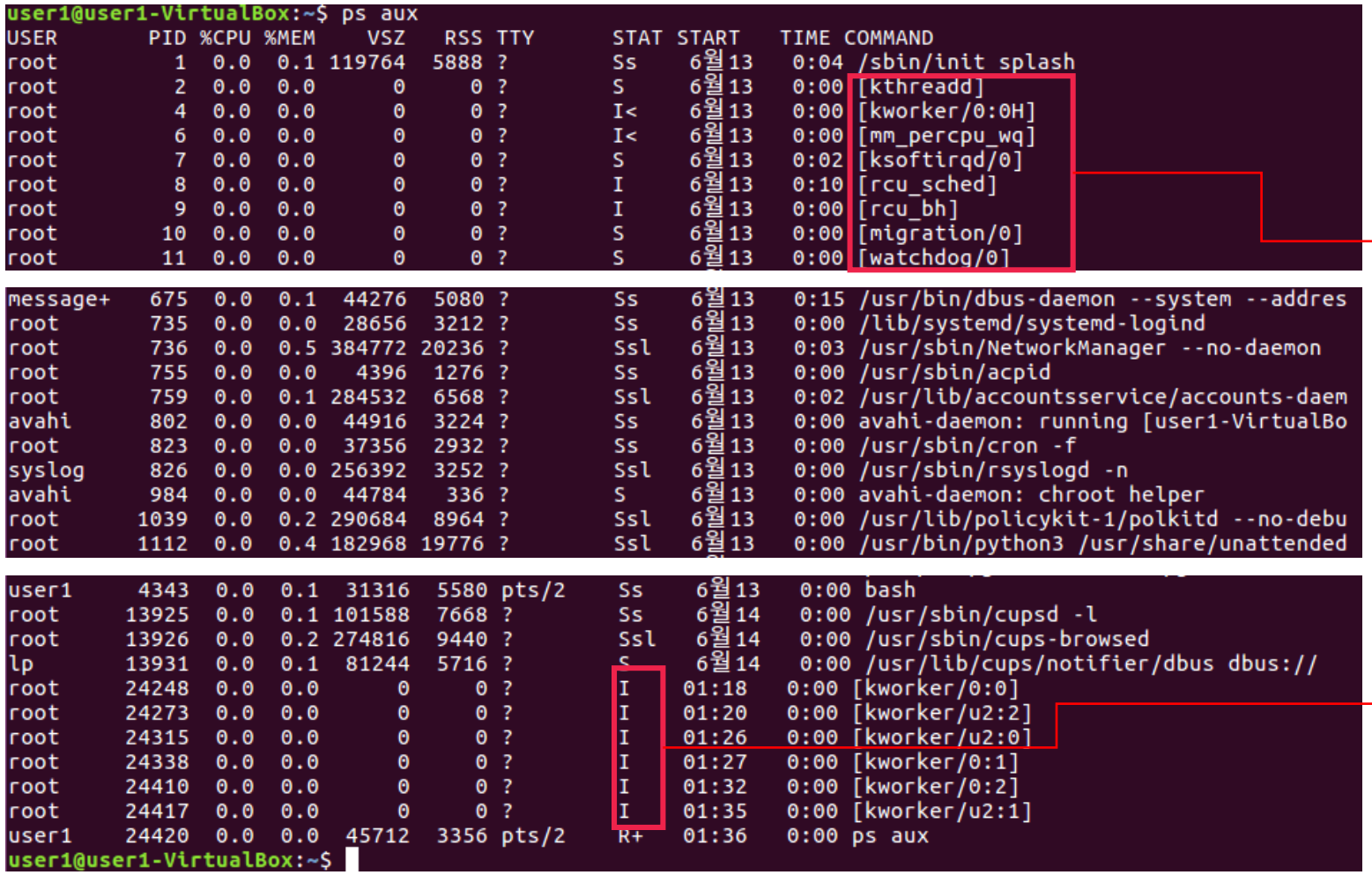
```
user1@user1-VirtualBox:~$ ps
  PID TTY          TIME CMD
10582 pts/1    00:00:00 bash
10706 pts/1    00:00:00 ps
user1@user1-VirtualBox:~$
```

```
user1@user1-VirtualBox:~$ ps -a
  PID TTY          TIME CMD
 2518 tty1     00:00:00 gnome-session-b
 2527 tty1     00:00:06 gnome-shell
 2560 tty1     00:00:00 Xwayland
 2611 tty1     00:00:00 ibus-daemon
 2614 tty1     00:00:00 ibus-dconf
 2617 tty1     00:00:00 ibus-x11
 2649 tty1     00:00:00 gsd-xsettings
 2661 tty1     00:00:00 gsd-a11y-settin
 2664 tty1     00:00:00 gsd-clipboard
```

```
user1@user1-VirtualBox:~$ ps a
  PID TTY      STAT   TIME COMMAND
 2514 tty1     Ssl+   0:00 /usr/lib/gdm3/gdm-wayland-session gnome-session
 2518 tty1     Sl+    0:00 /usr/lib/gnome-session/gnome-session-binary --au
 2527 tty1     Sl+    0:06 /usr/bin/gnome-shell
 2560 tty1     S+     0:00 /usr/bin/Xwayland :1024 -rootless -terminate -ac
 2611 tty1     Sl     0:00 ibus-daemon --xim --panel disable
 2614 tty1     Sl     0:00 /usr/lib/ibus/ibus-dconf
```

프로세스 확인 #2 - ps aux

ps - 프로세스 관리를 위한 all-in-one 유틸리티



- a : all processes
- u : show user/owner
- x : show all process

Kernel Task

Kernel Task 상태값
I : TASK_IDLE

프로세스 확인 #3 - ps 주요 옵션 조합

ps - 프로세스 디버깅을 위한 주요 명령어

- 모든 프로세스 살펴보기
  - ps ax
  - ps axu
- 프로세스 트리를 살펴보기
  - ps -ejH
  - ps axjf
- root 권한으로 실행중인 모든 프로세스 확인
  - ps -U root -u root u
    - 일반적으로는 UID 와 EUID 가 같으나, setuid bit이 설정되어 있는 경우 한시적으로 root 권한을 가져올 수 있음
- 내 권한으로 실행중인 모든 프로세스 확인
  - ps -xu
- 내가 원하는 필드만 출력
  - ps -eo pid,tid,class,rtprio,ni,pri,psr,pcpu,stat,wchat:14,comm
  - ps axo stat,euid,ruid,tty,tpgid,sess,pgrp,ppid,pid,pcpu,comm
  - ps -eo pid,tt,user,fname,tmout,f,wchan

- -U : real UID
- -u : effective UID

- ps -aN --format cmd,pid,user,ppid

프로세스 확인 - /proc 파일 시스템

/proc/<pid>/ - man 5 proc

- /proc/1 - init 프로세스의 정보
- /proc/self - 현재 실행중인 프로세스의 정보
  - eg, ls -l /proc/self : ls 프로세스 그 자체
  - cd /proc/self : bash 쉘 프로세스
    - cat cmdline : bash 프로세스의 실행 옵션 (bash)
- /proc/<pid>/
  - maps : 프로세스의 메모리 mapping 공간
  - cmdline : 프로세스 실행 인자
  - cwd : 프로세스가 사용중인 디렉토리나 파일
  - environ : 프로세스의 환경 변수
  - exe : 실행중인 프로그램의 이름
  - fd : 프로세스가 사용중인 파일 디스크럽터
  - fdinfo : 파일 디스크립터 정보
  - net : 프로세스가 바라보는 네트워크 정보
  - stat : 프로세스에 대한 정보 기록 (pid, name, status, ppid 등)
  - statm : 메모리 사용 정보 (size, resident, share, text, lib, data, dt)
  - status : 프로세스 상태 정보

프로세스 확인 - /proc 파일 시스템 #2 - 응용

nginx 웹서버 디버깅 - pidof nginx, cd /proc/<pid>

- cat cmdline
  - 내용 확인 및 systemctl 의 데몽 실행 명령어와 비교

```
user1@user1-VirtualBox:/proc/1222$ cat cmdline
nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
```

```
ExecStartPre=/usr/sbin/nginx -t -q -g 'daemon on; master_process on;'
ExecStart=/usr/sbin/nginx -g 'daemon on; master_process on;'
```

- sudo ls -al fd
  - 열고 있는 파일 디스크립터 확인

```
user1@user1-VirtualBox:/proc/1222$ sudo ls -al fd
합계 0
dr-x------ 2 root root  0  6월 21 23:03 .
dr-xr-xr-x 9 root root  0  6월 21 23:00 ..
lrwx------ 1 root root 64  6월 21 23:03 0 -> /dev/null
lrwx------ 1 root root 64  6월 21 23:03 1 -> /dev/null
l-wx------ 1 root root 64  6월 21 23:03 2 -> /var/log/nginx/error.log
lrwx------ 1 root root 64  6월 21 23:03 3 -> socket:[19166]
l-wx------ 1 root root 64  6월 21 23:03 4 -> /var/log/nginx/access.log
l-wx------ 1 root root 64  6월 21 23:03 5 -> /var/log/nginx/error.log
lrwx------ 1 root root 64  6월 21 23:03 6 -> socket:[19147]
lrwx------ 1 root root 64  6월 21 23:03 7 -> socket:[19148]
lrwx------ 1 root root 64  6월 21 23:03 8 -> socket:[19167]
```

- cat statm
  - 메모리 사용 정보 요약
- cat status
  - 프로세스의 상태 확인

```
user1@user1-VirtualBox:/proc/1222$ cat statm
31245 354 18 270 0 403 0
user1@user1-VirtualBox:/proc/1222$ cat status
Name:   nginx
Umask:  0000
State:  S (sleeping)
Tgid:   1222
Ngid:   0
Pid:    1222
PPid:   1
TracerPid:      0
Uid:    0       0       0       0
Gid:    0       0       0       0
FDSize: 64
Groups:
NStgid: 1222
NSpid:  1222
NSpgid: 1222
NSsid:  1222
VmPeak:   124980 kB
VmSize:   124980 kB
```

프로세스 트리

pstree 유틸리티

- pstree -u user1
  - 나(user1)의 권한으로 실행중인 프로세스
  - ps xf 와 동일한 결과

```
user1@user1-VirtualBox:~$ pstree
systemd─┬─ModemManager───2*[{ModemManager}]
        ├─NetworkManager─┬─dhclient
        │                └─2*[{NetworkManager}]
        ├─2*[VBoxClient───VBoxClient]
        ├─VBoxClient───VBoxClient───2*[{VBoxClient}]
        ├─VBoxClient───VBoxClient───3*[{VBoxClient}]
        ├─VBoxService───8*[{VBoxService}]
        ├─accounts-daemon───2*[{accounts-daemon}]
        ├─acpid
        ├─avahi-daemon───avahi-daemon
        ├─boltd───2*[{boltd}]
        ├─colord───2*[{colord}]
        ├─containerd───8*[{containerd}]
        ├─cron
        ├─cups-browsed───2*[{cups-browsed}]
        ├─cupsd
        ├─dbus-daemon
        ├─fwupd───4*[{fwupd}]
        ├─gdm3─┬─gdm-session-wor─┬─gdm-wayland-ses─┬─gnome-session-b─┬─gnome-sh+
        │      │                 │                 │                 ├─gsd-a11y+
        │      │                 │                 │                 ├─gsd-clip+
        │      │                 │                 │                 ├─gsd-colo+
        │      │                 │                 │                 ├─gsd-date+
```

```
user1@user1-VirtualBox:~$ pstree -u user1
VBoxClient───VBoxClient

VBoxClient───VBoxClient───2*[{VBoxClient}]

VBoxClient───VBoxClient───3*[{VBoxClient}]

gdm-x-session─┬─Xorg───{Xorg}
              ├─gnome-session-b─┬─deja-dup-monito───3*[{deja-dup-monito}]
              │                 ├─gnome-shell─┬─ibus-daemon─┬─ibus-dconf───3*[{ibus
              │                 │             │             ├─ibus-engine-han───2*[
              │                 │             │             └─2*[{ibus-daemon}]
              │                 │             └─9*[{gnome-shell}]
              │                 ├─gnome-software───3*[{gnome-software}]
              │                 ├─gsd-a11y-settin───3*[{gsd-a11y-settin}]
              │                 ├─gsd-clipboard───2*[{gsd-clipboard}]
```

프로세스 종료(?) - 특정 시그널 보내기

kill 명령어를 통한 프로세스 강제 종료 및 신호(시그널) 보내기

- 모든 시그널은 프로세스 구현자(개발자) 에게 그 개발여부(implementation) 가 달려 있음.
  - SIGHUP - 종료 (Hang-Up) 이지만, 주로 이를 통해 설정파일을 다시 불러오게도 쓰임
  - SIGQUIT - 정상적인 종료 후 코어덤프 생성
  - SIGTERM - 정상적인 종료
  - SIGINT - 강제 종료 (Ctrl+C)

- 구현자(개발자) 가 특별히 핸들링 할 수 없는 명령어
  - SIGKILL - 강제 종료

- 사용법
  - kill -HUP <pid>
  - kill -INT <pid>
  - kill -KILL <pid>

  - kill -1 <pid>
  - kill -2 <pid>
  - kill -9 <pid>

```
user1@user1-VirtualBox:~$ kill -l
 1) SIGHUP       2) SIGINT       3) SIGQUIT      4) SIGILL       5) SIGTRAP
 6) SIGABRT      7) SIGBUS       8) SIGFPE       9) SIGKILL     10) SIGUSR1
11) SIGSEGV     12) SIGUSR2     13) SIGPIPE     14) SIGALRM     15) SIGTERM
16) SIGSTKFLT   17) SIGCHLD     18) SIGCONT     19) SIGSTOP     20) SIGTSTP
21) SIGTTIN     22) SIGTTOU     23) SIGURG      24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM   27) SIGPROF     28) SIGWINCH    29) SIGIO       30) SIGPWR
31) SIGSYS      34) SIGRTMIN    35) SIGRTMIN+1  36) SIGRTMIN+2  37) SIGRTMIN+3
38) SIGRTMIN+4  39) SIGRTMIN+5  40) SIGRTMIN+6  41) SIGRTMIN+7  42) SIGRTMIN+8
43) SIGRTMIN+9  44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7
58) SIGRTMAX-6  59) SIGRTMAX-5  60) SIGRTMAX-4  61) SIGRTMAX-3  62) SIGRTMAX-2
63) SIGRTMAX-1  64) SIGRTMAX
```

프로세스 종료(?) - 특정 시그널 보내기

killall 명령어를 통한 프로세스 종료(SIGTERM) 및 신호(시그널) 보내기

- 사용법
  - killall <프로세스명>
  - killall -s <신호> <프로세스명>
  - killall -u <사용자> <프로세스명>

- 실제 사용 예
  - sudo killall nginx

  - killall -i -v bash
    - -i : interactive (종료시 확인)
    - -v : verbose (실행 결과를 보여줌)

```
user1@user1-VirtualBox:~$ killall -i -v bash
Kill bash(10547) ? (y/N) y
Killed bash(10547) with signal 15
Kill bash(10582) ? (y/N) N
user1@user1-VirtualBox:~$
```

프로세스 디버깅 - lsof (list open files)

lsof - 파일을 사용중인 프로세스 조회

- lsof /path/to/binary
  - 특정 프로세스가 사용주인 파일시스템 확인

```
user1@user1-VirtualBox:~$ sudo lsof /usr/sbin/nginx
lsof: WARNING: can't stat() fuse.gvfsd-fuse file system /run/user/1000/gvfs
      Output information may be incomplete.
COMMAND  PID     USER  FD   TYPE DEVICE SIZE/OFF NODE NAME
nginx   1384     root txt    REG    8,1  1149096  750 /usr/sbin/nginx
nginx   1386 www-data txt    REG    8,1  1149096  750 /usr/sbin/nginx
user1@user1-VirtualBox:~$
```

- lsof -i
  - 네트워크를 이용중인 프로세스 조회

- lsof -u <username>
  - 특정 사용자가 실행한 프로세스가 사용중인 파일

```
user1@user1-VirtualBox:~$ sudo lsof -i
COMMAND    PID            USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
systemd-r  404 systemd-resolve   12u  IPv4  14872      0t0  UDP localhost:domain
systemd-r  404 systemd-resolve   13u  IPv4  14873      0t0  TCP localhost:domain (LISTEN)
avahi-dae  761           avahi   12u  IPv4  18989      0t0  UDP *:mdns
avahi-dae  761           avahi   13u  IPv6  18990      0t0  UDP *:mdns
avahi-dae  761           avahi   14u  IPv4  18991      0t0  UDP *:34135
avahi-dae  761           avahi   15u  IPv6  18992      0t0  UDP *:57399
vsftpd    1287            root    3u  IPv6  21600      0t0  TCP *:ftp (LISTEN)
container 1306            root    8u  IPv4  24672      0t0  TCP localhost:45003 (LISTEN)
sshd      1366            root    3u  IPv4  24243      0t0  TCP *:ssh (LISTEN)
sshd      1366            root    4u  IPv6  24247      0t0  TCP *:ssh (LISTEN)
dhclient  1376            root    6u  IPv4  22161      0t0  UDP *:bootpc
nginx     1384            root    6u  IPv4  22101      0t0  TCP *:http (LISTEN)
nginx     1384            root    7u  IPv6  22102      0t0  TCP *:http (LISTEN)
nginx     1386        www-data    6u  IPv4  22101      0t0  TCP *:http (LISTEN)
nginx     1386        www-data    7u  IPv6  22102      0t0  TCP *:http (LISTEN)
postgres  1767        postgres    5u  IPv4  24480      0t0  TCP localhost:postgresql (LISTEN)
postgres  1767        postgres    9u  IPv4  24636      0t0  UDP localhost:60159->localhost:60159
```

프로세스 디버깅 - fuser

fuser - 파일/디렉토리를 사용중인 프로세스/사용자 조회

- 사용법
  - fuser /path/to/file

- 실 사용 예시
  - sudo fuser /var/log/nginx/*
    - 엔진엑스 아래 로그 파일을 사용중인 프로세스 목록
  - sudo fuser /var/log/*
    - 로그파일을 사용중인 프로스스 목록

```
user1@user1-VirtualBox:~$ sudo fuser /var/log/nginx/*
/var/log/nginx/access.log:  1230  1231
/var/log/nginx/error.log:  1230  1231
user1@user1-VirtualBox:~$
user1@user1-VirtualBox:~$ sudo fuser /var/log/*
/var/log/Xorg.0.log:  2801
/var/log/auth.log:     826
/var/log/kern.log:     826
/var/log/syslog:       826
user1@user1-VirtualBox:~$
```

프로세스 디버깅 - strace

strace - 시스템 콜 트레이싱

- strace <cmd>

- strace -t <cmd>
  - 출력 결과에 timestamp 표시
- strace -tt <cmd>
  - 출력 결과에 timestamp.msec 표시

- strace -f <cmd>
  - fork 프로세스까지 추적

```
user1@user1-VirtualBox:~$ strace whoami
execve("/usr/bin/whoami", ["whoami"], [/* 64 vars */]) = 0
brk(NULL)                               = 0x21fd000
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=88915, ...}) = 0
mmap(NULL, 88915, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f828b7d7000
close(3)                                = 0
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
open("/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0P\t\2\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0755, st_size=1868984, ...}) = 0
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f828b7d6000
mmap(NULL, 3971488, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f828b1fe000
mprotect(0x7f828b3be000, 2097152, PROT_NONE) = 0
mmap(0x7f828b5be000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1c0000) = 0x7
f828b5be000
```

```
user1@user1-VirtualBox:~$ strace ls
execve("/bin/ls", ["ls"], [/* 64 vars */]) = 0
brk(NULL)                               = 0x18e8000
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=88915, ...}) = 0
mmap(NULL, 88915, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f5b8c7be000
close(3)                                = 0
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
open("/lib/x86_64-linux-gnu/libselinux.so.1", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\260Z\0\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0644, st_size=130224, ...}) = 0
```

```
write(1, "ddd\t   dir3\t\t     find_old.sh    "..., 77ddd          dir3              find_old.sh    hello
symlink  test1.sh       다운로드  사진
) = 77
write(1, "dir.txt    examples.desktop  fin"..., 77dir.txt    examples.desktop  find_opt.sh    mycat
 test2.sh       문서      음악
) = 77
close(1)                                = 0
close(2)                                = 0
exit_group(0)                           = ?
+++ exited with 0 +++
user1@user1-VirtualBox:~$
```

프로세스 디버깅 - ltrace

ltrace - 라이브러리 콜 트레이싱

```
user1@user1-VirtualBox:~$ ltrace ls
malloc(552)                                                    = 0x1953010
malloc(120)                                                    = 0x1953240
malloc(1024)                                                   = 0x19532c0
free(0x19532c0)                                                = <void>
free(0x1953010)                                                = <void>
__libc_start_main(0x402a00, 1, 0x7ffcf1f450c8, 0x413bb0 <unfinished ...>
strrchr("ls", '/')                                             = nil
setlocale(LC_ALL, "" <unfinished ...>
malloc(5)                                                      = 0x1953010
free(0x1953010)                                                = <void>
malloc(120)                                                    = 0x1953030
malloc(12)                                                     = 0x1953010
```

```
memcpy(0x1958d18, "\020H\225\001\0\0\0\0", 8)                  = 0x1958d18
__errno_location()                                             = 0x7f07052656b8
strcoll("\353\254\270\354\204\234", "readonly.txt")            = 113
__errno_location()                                             = 0x7f07052656b8
strcoll("hellolink", "test2.sh")                               = -12
memcpy(0x1958da0, "PD\225\001\0\0\0\0", 8)                     = 0x1958da0
__errno_location()                                             = 0x7f07052656b8
strcoll("hellolink", "readonly.txt")                           = -10
__errno_location()                                             = 0x7f07052656b8
```

```
fclose(0x7f0704e34620 <unfinished ...>
free(0x195a870)                                                = <void>
<... fclose resumed> )                                         = 0
__fpending(0x7f0704e34540, 0, 0x7f0704e35780, 0)               = 0
fileno(0x7f0704e34540)                                         = 2
__freading(0x7f0704e34540, 0, 0x7f0704e35780, 0)               = 0
__freading(0x7f0704e34540, 0, 4, 0)                            = 0
fflush(0x7f0704e34540)                                         = 0
fclose(0x7f0704e34540)                                         = 0
+++ exited (status 0) +++
user1@user1-VirtualBox:~$
```