Fault injection
テストのコードカバレッジを上げる

Yoichi Yuasa

OSAKA NDS Embedded Linux Cross Forum #10

自己紹介

- 湯浅陽一
- 1999年よりLinux kernel開発に参加
- MIPSアーキテクチャのいくつかのCPUへLinux kernelを移植

本日の内容

- Fault Injectionとは
- Linux Fault-injectionの紹介
- Linux Fault-injectionの組込み

Fault Injectionとは

- エラー処理など通常は発生しない状況をテストするために意図的に状態を発生させる
  - ハードウエアを改造して状態を発生させる
  - ソフトウエアを変更して状態を発生させる
- テストカバレッジを上げるための手段

Linux Fault-injection

- Fault injectionのためのフレームワークと各機能向けの個別実装
  - kmalloc
  - alloc_pages
  - Disk I/O
  - futex
  - MMC I/Oなど

Fault-injection kernel configuration menu

Kernel hacking → Kernel Testing and Coverage

[*] Fault-injection framework

[*] Fault-injection capability for kmalloc

[*] Fault-injection capability for alloc_pages()

[*] Fault-injection capability for disk IO

[*] Fault-injection capability for faking disk interrupts

[*] Fault-injection capability for futexes

[*] Debugfs entries for fault-injection capabilities

[*] Fault-injection capability for MMC IO

Linux Fault-injectionの仕組み

- 処理の中にshould_fail関数を追加して、返り値がtrueのときは発生させたい状態に遷移させる
- falseのときは通常の処理

Linux Fault-injection

- debugfs上のファイルインターフェースで制御
- 発生確率、間隔や回数など制御可能
- debugfsから設定した値をshould_fail関数内で判定

インターフェース

- probability
- interval
- times
- space
- verbose
- verbose_ratelimit_interval_ms
- verbose_ratelimit_burst
- task-filter

probability

- should_fail関数がtrueとなる確率
- パーセントで設定(0〜100)

interval

- should_fail関数がtrueになる間隔
- 2回に1回なら2に設定
- 正確にintervalでtrueにするにはprobabilityを100に設定

times

- should_fail関数がtrueになる回数
- -1は無限
- should_fail関数がtrueになると1減算される

space

- should_fail関数の第2引数であるsizeで指定した数だけ減算されていく
- spaceがsize以下になるとshould_fail関数がtrueを返すようになる
- 少し経過してからFault-injectionを機能させたい場合に利用

verbose

- should_fail関数がtureのときに出力されるメッセージの内容を設定
  - 0: 無出力
  - 1: FAULT INJECTION: forcing a failure.で始まる文字列出力、出力レベルはKERN_NOTICE
  - 2: 1の出力+スタックダンプ

verbose_ratelimit_burst

- verboseのメッセージを設定した回数までしか表示しないようにする
- 何度も同じメッセージを表示しないように設定するために利用

verbose_ratelimit_interval_ms

- verbose_ratelimit_burstの計算期間を設定(jiffies)
- 設定期間を経過するとburstと比較する値をリセット
- interval_msが0の場合はburst設定は無効

task-filter

- should_fail関数が機能するかをPID毎に設定
  - N: 無効
  - Y: 有効
- /proc/<pid>/make-it-failを1に設定しているとそのPIDでは他の条件に従ってshould_fail関数がtrueになる

Linux Fault-injectionの組込み

- Kernel configurationを有効にする
  - [*] Fault-injection framework
  - [*]   Debugfs entries for fault-injection capabilities
- struct fault_attrを定義する
- should_fail関数を追加する
- 実際にI2C read/writeに組込んでみる

struct fault_attrの定義

- 利用しているI2Cホストコントローラドライバへ以下を追加

  include <linux/fault-inject.h>

  DECLARE_FAULT_ATTR(fail_i2c_attr);
- DECLARE_FAULT_ATTRマクロではprobabilityが0
- 起動時から機能させたい場合は個別に設定
- interval,probability,space,timesはkernel command lineから設定することも可能

struct fault_attrの定義

```
struct fault_attr fail_i2c_attr = {
    .probability = 100,
    .interval = 1,
    .times = ATOMIC_INIT(10),
    .space = ATOMIC_INIT(0),
    .verbose = 2,
    .task_filter = false,
    .ratelimit_state = RATELIMIT_STATE_INIT_DISABLED,
    .dname = NULL,
};
```

should_fail関数

bool should_fail(struct fault_attr *attr, ssize_t size)

- sizeはshould_fail関数呼び出し時にspaceから減算される値

should_fail関数の追加

- I2C read/writeで通る__i2c_transfer関数にshould_fail関数を追加
- __i2c_transfer関数の返り値をshould_fail関数がtrueのときに書き換える

```
if (shoud_fail(&fail_i2c_attr, 1) {
        static const int i2c_errors[] = {
                -ETIMEDOUT,
                -ENXIO,
                -EBUSY,
        };
        ret = i2c_errors[prandom_u32() % ARRAY_SIZE(i2c_errors)];
}
return ret;
```

Fault-injection発生

```
/sys/kernel/debug/fail_i2c# ls -1
interval
probability
space
task-filter
times
verbose
verbose_ratelimit_burst
verbose_ratelimit_interval_ms
/sys/kernel/debug/fail_i2c# ls -1 | xargs cat
1
100
0
N
8 ← 10に設定したtimesが2回Fault-injectionが発生したことにより減算されている
2
10
0
```

Fault-injection時メッセージ

```
FAULT_INJECTION: forcing a failure.
name fail_i2c, interval 1, probability 100, space 0, times 9
CPU: 0 PID: 1 Comm: swapper/0 Not tainted 4.14.162 #14
Hardware name: ????
Call trace:
[<ffff00000808966c>] dump_backtrace+0x0/0x390
[<ffff000008089a10>] show_stack+0x14/0x1c
[<ffff000008778054>] dump_stack+0xc0/0x100
[<ffff0000083aef48>] should_fail+0x80/0x184
[<ffff000008544a0c>] __i2c_transfer+0x41c/0x4e4
[<ffff000008544b50>] i2c_transfer+0x7c/0xb0
[<ffff000008544bd8>] i2c_transfer_buffer_flags+0x54/0x7c
[<ffff0000084c9eec>] regmap_i2c_write+0x1c/0x44
[<ffff0000084c6178>] _regmap_raw_write+0x51c/0x714
[<ffff0000084c63d8>] _regmap_bus_raw_write+0x68/0x74
[<ffff0000084c5158>] _regmap_write+0xdc/0x150
[<ffff0000084c6670>] regmap_write+0x48/0x70
…
0-0058: i2c write error ret=-110
```

Fault-injectionまとめ

- 少ない変更で動作状態を動的に変更できるようになる
- 簡易なインターフェースで柔軟に発生条件を設定できる
- エラー処理がきちんとチェックできる

参考

- https://www.static.linuxfound.org/jp_uploads/seminar20070710/LinuxFaultInjection-2.pdf
- Linux kernel source code
  - Documentation/fault-injection/fault-injection.txt