

# perf/uftrace 컨트리뷰톤 2017

2017.11. 21 (화)

KOSSCON 2017

컨트리뷰톤 성과발표회

발표자: 홍한석

# 목차

- 프로젝트 및 팀원 소개
- 개발 과정
- 성과
- Features
- 느낀점

# 프로젝트 소개

# perf

- Linux profiling with performance analyzer
- A very active subproject of Linux Kernel;  
**50+patches/week !!**
- C/C++, Assembly

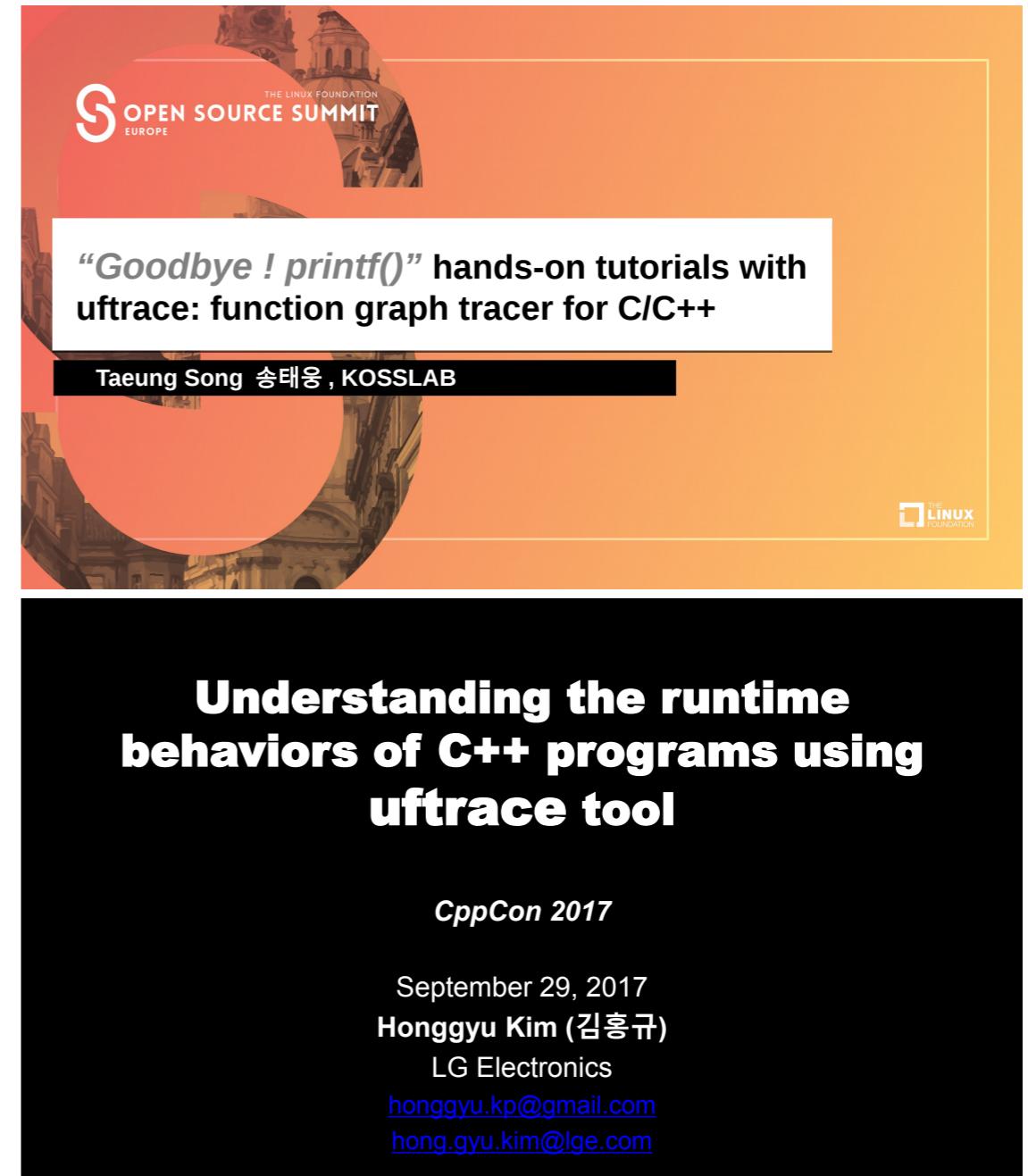
```
[kernel.kallsyms] with build id 13a40a2e59484522176789a1ae4b311dfd75
# To display the perf.data header info, please use --header/--header
#
# Total Lost Samples: 0
#
# Samples: 68 of event 'cpu-clock'
# Event count (approx.): 17000000
#
# Overhead  Command           Shared Object      Symbol
# .....,.
# 83.82%  old_pack_knapsa  old_pack_knapsack  [.] get_cond_maxprice
# 11.76%  old_pack_knapsa  old_pack_knapsack  [.] pack_knapsack
# 2.94%   old_pack_knapsa  [kernel.kallsyms]  [k] 0xfffffffffb1eb0d7
# 1.47%   old_pack_knapsa  ld-2.23.so       [.] dl_main

#
# (Tip: Sample related events with: perf record -e '{cycles,instruc
#
```

```
List of pre-defined events (to be used in -e):
cycles-ct OR cpu/cycles-ct/                                [Kernel PMU event]
cycles-t OR cpu/cycles-t/                                 [Kernel PMU event]
el-abort OR cpu/el-abort/                                 [Kernel PMU event]
el-capacity OR cpu/el-capacity/                            [Kernel PMU event]
el-commit OR cpu/el-commit/                               [Kernel PMU event]
el-conflict OR cpu/el-conflict/                            [Kernel PMU event]
el-start OR cpu/el-start/                                 [Kernel PMU event]
mem-loads OR cpu/mem-loads/                               [Kernel PMU event]
mem-stores OR cpu/mem-stores/                             [Kernel PMU event]
nsr/aperf/                                                 [Kernel PMU event]
nsr/mperf/                                                 [Kernel PMU event]
nsr/smi/                                                 [Kernel PMU event]
nsr/tsc/                                                 [Kernel PMU event]
power/energy-cores/                                       [Kernel PMU event]
power/energy-gpu/                                         [Kernel PMU event]
power/energy-pkg/                                         [Kernel PMU event]
power/energy-ran/                                         [Kernel PMU event]
ref-cycles OR cpu/ref-cycles/                             [Kernel PMU event]
topdown-fetch-bubbles OR cpu/topdown-fetch-bubbles/ [Kernel PMU event]
topdown-recovery-bubbles OR cpu/topdown-recovery-bubbles/ [Kernel PMU event]
topdown-slots-issued OR cpu/topdown-slots-issued/ [Kernel PMU event]
```

# uftrace

- Function (graph) tracer for user-space
- C/C++ 프로그램 실행흐름을 추적, 분석 가능한 명령줄 도구
- C, Python
- 발표: OSSNA & OSSEU 2017, CppCon 2016/2017, ...
- Star: 400 → 600+ 해외에서 많은 인기



# git like!; perf/uftrace

#<command> <sub-command> ...

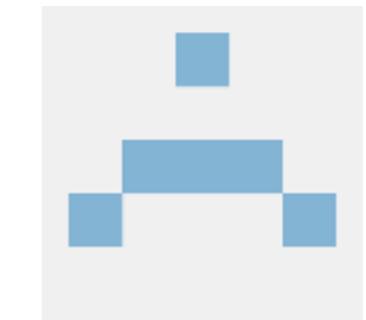
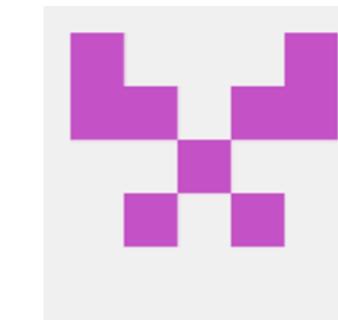
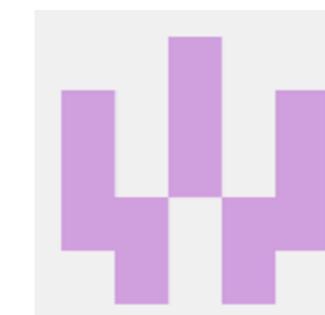
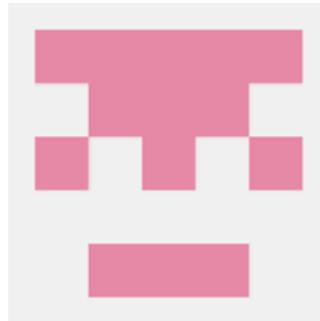
**git commit** -sm "this is commit message"

**perf record** -e block:block\_rq\_issue -ag

**uftrace record** -A foo@arg1 hello

# 팀원 소개

# perf/uftrace 팀



개발자: 학생, 직장인, 교수 ... → 오픈소스 개발자!

# 개발 과정

# 일정 관리 및 미션 수행

- 구글 드라이브 활용
  - 일정 공지 및 출석
  - 마일스톤 양식의 미션 수행표
  - 공부/참고 문서 및 자료 공유
- 소통
  - 단톡방/슬랙/기타

# 미션 수행표

|        | 컨트리뷰튼 2017 perf / utrace 미션 |               |               |                |                  |           |           |                 |          |              |
|--------|-----------------------------|---------------|---------------|----------------|------------------|-----------|-----------|-----------------|----------|--------------|
| 이름     | 1) Git Basic 실습             | 2) Git Adv 실습 | 3) Node.js 실습 | 4) perf 컵타일/실행 | 5) utrace 컵파일/실행 | 6) 플렉소개문서 | 7) 개발환경문서 | 8) commit 분석보고서 | 9) 다양한실습 | 10) ELF 문서실습 |
| 1 김자은  |                             |               |               |                |                  |           |           |                 |          |              |
| 2 조영식  |                             |               |               |                |                  |           |           |                 |          |              |
| 3 박한빈  |                             |               |               |                |                  |           |           |                 |          |              |
| 4 홍상민  |                             |               |               |                |                  |           |           |                 |          |              |
| 5 김기혁  |                             |               |               |                |                  |           |           |                 |          |              |
| 6 김유진  |                             |               |               |                |                  |           |           |                 |          |              |
| 7 전시현  |                             |               |               |                |                  |           |           |                 |          |              |
| 8 홍한석  |                             |               |               |                |                  |           |           |                 |          |              |
| 9 박성현  |                             |               |               |                |                  |           |           |                 |          |              |
| 10 이주신 |                             |               |               |                |                  |           |           |                 |          |              |
| 11 장소진 |                             |               |               |                |                  |           |           |                 |          |              |
| 12 김태성 |                             |               |               |                |                  |           |           |                 |          |              |
| 13 천관수 |                             |               |               |                |                  |           |           |                 |          |              |
| 14 원민체 |                             |               |               |                |                  |           |           |                 |          |              |
| 15 양혜임 |                             |               |               |                |                  |           |           |                 |          |              |
| 16 장혜정 |                             |               |               |                |                  |           |           |                 |          |              |
| 17 이창수 |                             |               |               |                |                  |           |           |                 |          |              |
| 18 고명진 |                             |               |               |                |                  |           |           |                 |          |              |
| 19 권기범 |                             |               |               |                |                  |           |           |                 |          |              |

- Git 익히기 / 문서화
- 개인 별 원하는 프로젝트(perf/uftrace)와 역량을 고려해 참여

# 문서화 및 보고서 작성 훈련

- 프로젝트 소개문서

- 프로젝트 파악에 도움이 되고 추후 더 많은 컨트리뷰션을 위함

- 개발환경 구성문서

- Commit 분석 보고서

- Commit 메세지나 코드 스타일, git blame으로 추적하여 컨트리뷰션에 대한 insight 도출

- ELF/PLT hooking 등 프로젝트에 대한 전문지식 공부 및 실습

- Github Issues 활용

Y kosslab-kr / uftrace  
forked from namhyung/uftrace

Code Issues 67 Pull requests 0

Filters is:issue is:open

67 Open ✓ 2 Closed

- [김유정] 프로젝트 소개문서 #43 opened on 14 Oct by yujeongKim
- [장소진] 커밋 분석 보고서 #42 opened on 14 Oct by JangSoJin
- [미향수] 커밋 분석 보고서 #41 opened on 14 Oct by aChangSuLee
- [임민체] 커밋 분석 보고서 #40 opened on 14 Oct by minchae
- [양혜임] 커밋 분석 보고서 #39 opened on 14 Oct by yanghyeimm
- [장시현] 커밋 분석 보고서 #38 opened on 14 Oct by sihyeon
- [장혜정] 커밋 분석 보고서 #37 opened on 14 Oct by love2986
- [장혜정] 개발환경 구성문서 #36 opened on 14 Oct by love2986
- [장혜정] 프로젝트 소개서 #35 opened on 14 Oct by love2986

# 프로젝트 소개문서



seonghyun0p commented on 14 Oct • edited

**uftrace**

Linux 환경에서 C/C++ 로 작성된 프로그램의 실행을 추적하고 분석하는 도구

## Features

함수 entry와 exit만 추적이 가능.  
필터 기능을 통해 함수를 포함, 또는 제외 가능.  
멀티 프로세스, 멀티 스레드 및 커널 함수에 대한 추적도 지원.

## Language

C, Python, etc.

```
cloc uftrace/
```

```
100 files
200 files
300 files
370 text files.
classified 367 files
Duplicate file check 367 files (337 known unique)
Unique:    100 files
Unique:    200 files
Unique:    300 files
```

# 개발환경 구성문서



sihyeonn commented on 14 Oct • edited ▾



## Environment

- OS : Linux mint 18.2 sonya Mate
- Kernel : 4.10.0-37-generic

## Uftrace Setting

1. Fork your own copy of [official uftrace](#) to your account

2. Get source files of uftrace

```
$ git clone https://github.com/<your git user name>/uftrace.git
```

3. Get library and make install

```
$ sudo apt-get install libelf-dev  
$ cd uftrace  
$ make  
$ make install
```

# Commit 분석 보고서



rjs1197 commented on 14 Oct • edited



## 1. kernel: Protect from crash on old kernel data

tests git:(master) ✘ git show 2e5f0b0

```
commit 2e5f0b02567c77fa9eb5f709292420cef2ff3be7
Author: Namhyung Kim <namhyung@gmail.com>
Date:   Wed Jun 21 19:40:57 2017 +0900

    kernel: Protect from crash on old kernel data

The old data doesn't contain required kernel metadata files. In this
case, utrace tries to access current system files but it failed due to
the permission. But it still has a pevent pointer which leads to an
invalid access to not initialized kernel data.

Signed-off-by: Namhyung Kim <namhyung@gmail.com>

diff --git a/utils/kernel.c b/utils/kernel.c
index 2c46fd7..9f8bf56 100644
--- a/utils/kernel.c
+++ b/utils/kernel.c
@@ -927,13 +927,13 @@ int setup_kernel_data(struct utrace_kernel *kernel)

    kernel->n_r_cpus = scandir(kernel->output_dir, &list, scandir_filter, versionsort);
    if (kernel->n_r_cpus <= 0) {
-        pr_log("cannot find kernel trace data\n");
-        return -1;
+        pr_out("cannot find kernel trace data\n");
+        goto out;
    }

    if (load_kernel_files(kernel) < 0) {
        pr_out("cannot read kernel header: %m\n");
-        return -1;
+        goto out;
    }

    pr_dbg("found kernel ftrace data for %d cpus\n", kernel->n_r_cpus);
@@ -989,6 +989,7 @@ int setup_kernel_data(struct utrace_kernel *kernel)
    free(list);
    if (i != kernel->n_r_cpus) {
        pr_dbg('failed to access to kernel trace data: %s: %m\n', buf);
-        finish_kernel_data(kernel);
+        return -1;
    }
}
```

# 참고자료 공유

|             | 참고자료 / 도서  | 링크  |
|-------------|--|---|
| 개막식 (9/28)  | 1 리눅스 커널 소스 (perf, ftrace, ...)                        | <a href="https://git.kernel.org/pub/scm/linux/kernel/git/tip/tip.git">https://git.kernel.org/pub/scm/linux/kernel/git/tip/tip.git</a>                                       |
|             | 2 Uftrace 소스코드: Function (graph) tracer for user-space | <a href="https://github.com/namhyung/uftrace">https://github.com/namhyung/uftrace</a>   |
|             | 3 Contributhon 전체적인 과정(Stage 1~5) PDF                  | <a href="https://www.dropbox.com/s/qv07u574q0qbp9j/Contributhon.pdf?dl=1">https://www.dropbox.com/s/qv07u574q0qbp9j/Contributhon.pdf?dl=1</a>                               |
|             | 4 uftrace 공식 채팅 gitter                                 | <a href="https://gitter.im/uftrace/uftrace">https://gitter.im/uftrace/uftrace</a>   |
|             | uftrace 소개 영상  | <a href="https://www.youtube.com/watch?v=LNav5qvyK7I">https://www.youtube.com/watch?v=LNav5qvyK7I</a>   |
| 1회차 (10/11) | 5 Git 훈련 (간단프로그램 시나리오 베이스)                             | <a href="https://github.com/taeung/git-training">https://github.com/taeung/git-training</a>   |
|             | 6 튜토리얼 1: Git의 기본                                      | <a href="https://backlogtool.com/git-guide/kr/intro/intro2_1.html">https://backlogtool.com/git-guide/kr/intro/intro2_1.html</a>   |
|             | 7 Try Git (Git명령실습)                                    | <a href="https://try.github.io/levels/1/challenges/1">https://try.github.io/levels/1/challenges/1</a>   |
|             | 8 Git 브랜치 배우기 실습                                       | <a href="http://learnbranch.urigit.com/">http://learnbranch.urigit.com/</a>   |
|             | 9 Advanced Git 실습자료                                    | <a href="https://www.dropbox.com/s/bkyptatfzr7e7v/Advanced_Git.pdf?dl=1">https://www.dropbox.com/s/bkyptatfzr7e7v/Advanced_Git.pdf?dl=1</a>                                 |
|             | 10   |   |
|             | 11 프로세스 메모리 구조 (간단하게 소스코드와 매칭)                         | <a href="http://skensita.tistory.com/entry/%EB%A9%94%EB%AA%A8%EB%A6%AC%EA%B5%AC">http://skensita.tistory.com/entry/%EB%A9%94%EB%AA%A8%EB%A6%AC%EA%B5%AC</a>                 |
|             | 12 프로세스 메모리 구조 (영역별 설명)                                | <a href="http://sfixer.tistory.com/entry/%EB%A9%94%EB%AA%A8%EB%A6%AC-%EC%98%81%EB%A9%A0">http://sfixer.tistory.com/entry/%EB%A9%94%EB%AA%A8%EB%A6%AC-%EC%98%81%EB%A9%A0</a> |
|             | 13 뇌를자극하는 프로그래밍원리 chapter 4: 컴퓨터의 두뇌 - CPU             | <a href="http://cfile223.uf.daum.net/attach/1626080B4A60B218071E7A">http://cfile223.uf.daum.net/attach/1626080B4A60B218071E7A</a>   |
|             | 14 뇌를자극하는 프로그래밍원리 chapter 13: 함수가호출되기까지                | <a href="http://cfile220.uf.daum.net/attach/1826130B4A60B21B039DF5">http://cfile220.uf.daum.net/attach/1826130B4A60B21B039DF5</a>   |
|             | 15 uftrace youtube 6분영상                                | <a href="https://youtu.be/LNav5qvyK7I">https://youtu.be/LNav5qvyK7I</a>   |
|             | 16 ELF 스터디 자료  | <a href="https://www.dropbox.com/s/f8ikkl6kocyhsfx/elf.pdf?dl=1&amp;pl=1">https://www.dropbox.com/s/f8ikkl6kocyhsfx/elf.pdf?dl=1&amp;pl=1</a>                               |
|             | 17 uftrace 기술 문서 (미완성)                                 | <a href="https://www.dropbox.com/s/cppqy79yiwbadh3/uftrace.pdf?dl=1">https://www.dropbox.com/s/cppqy79yiwbadh3/uftrace.pdf?dl=1</a>   |
|             | 18 GDB 명령어 정리  | <a href="http://mintnlatte.tistory.com/">http://mintnlatte.tistory.com/</a>   |
|             | 19 ELF format 공식 문서                                    | <a href="http://www.skyfree.org/linux/references/ELF_Format.pdf">http://www.skyfree.org/linux/references/ELF_Format.pdf</a>   |
|             | 20 PLT와 GOT 자세히 알기 1                                   | <a href="https://bpsecblog.wordpress.com/2016/03/07/about_got_plt_1/">https://bpsecblog.wordpress.com/2016/03/07/about_got_plt_1/</a>                                       |
|             | 21 PLT와 GOT 자세히 알기 2                                   | <a href="https://bpsecblog.wordpress.com/2016/03/09/about_got_plt_2/">https://bpsecblog.wordpress.com/2016/03/09/about_got_plt_2/</a>                                       |

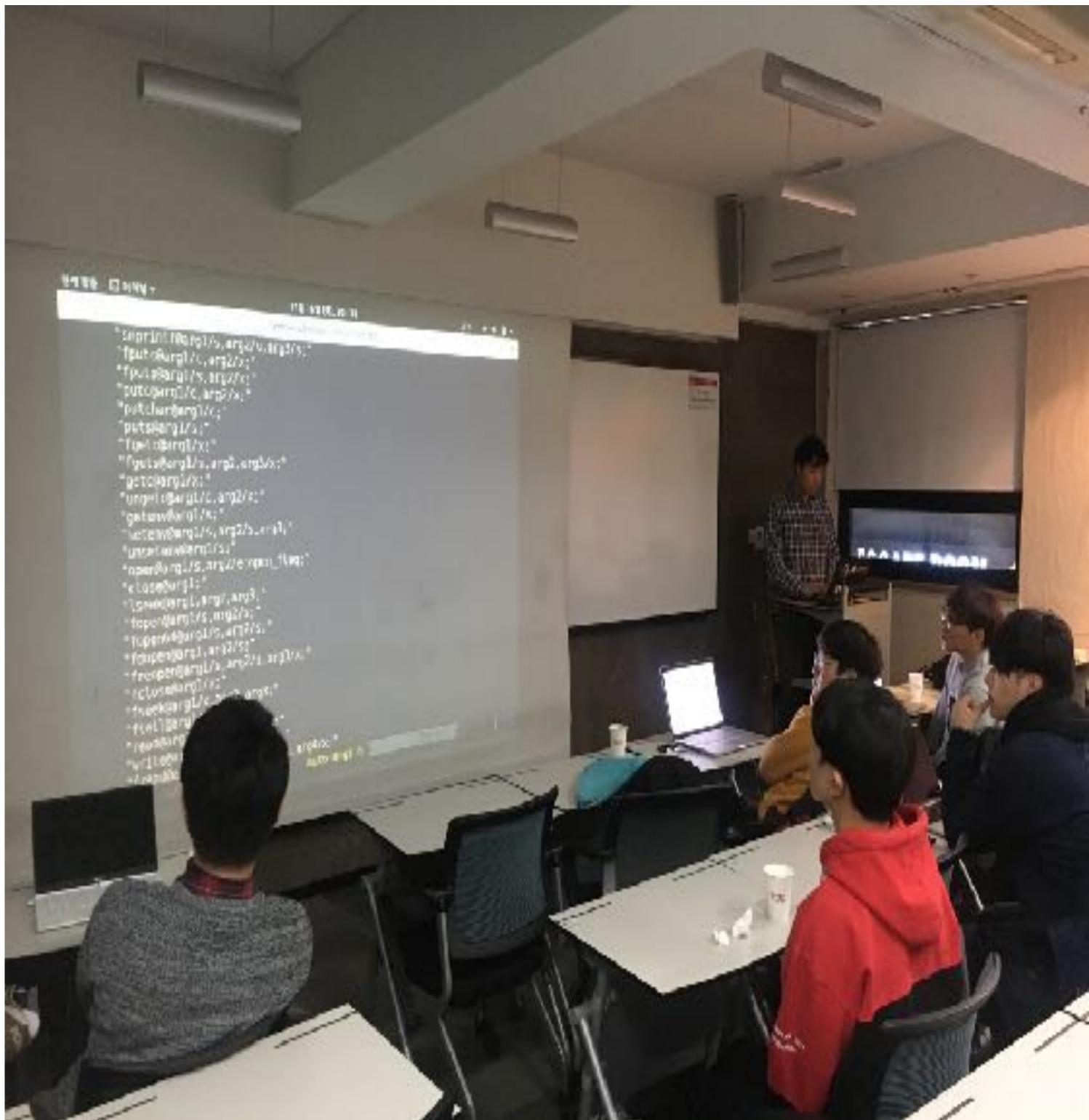
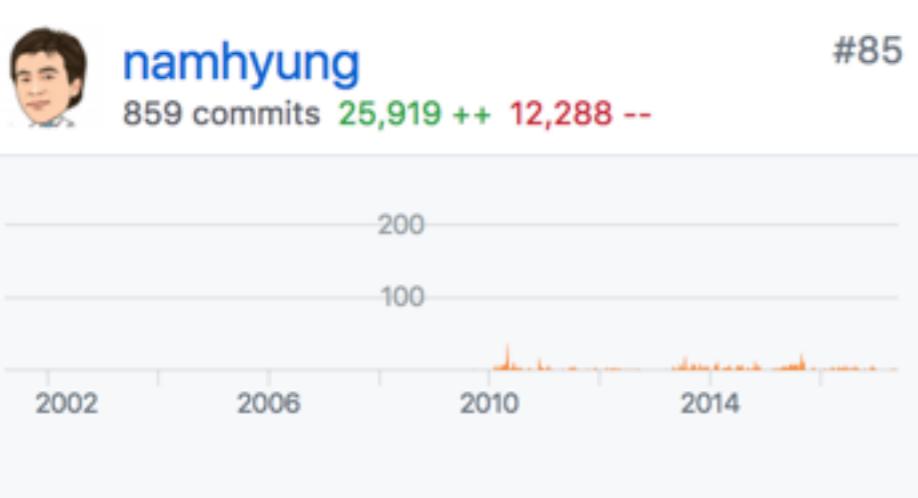
- 컨트리뷰션!!

# 멘토링 지도

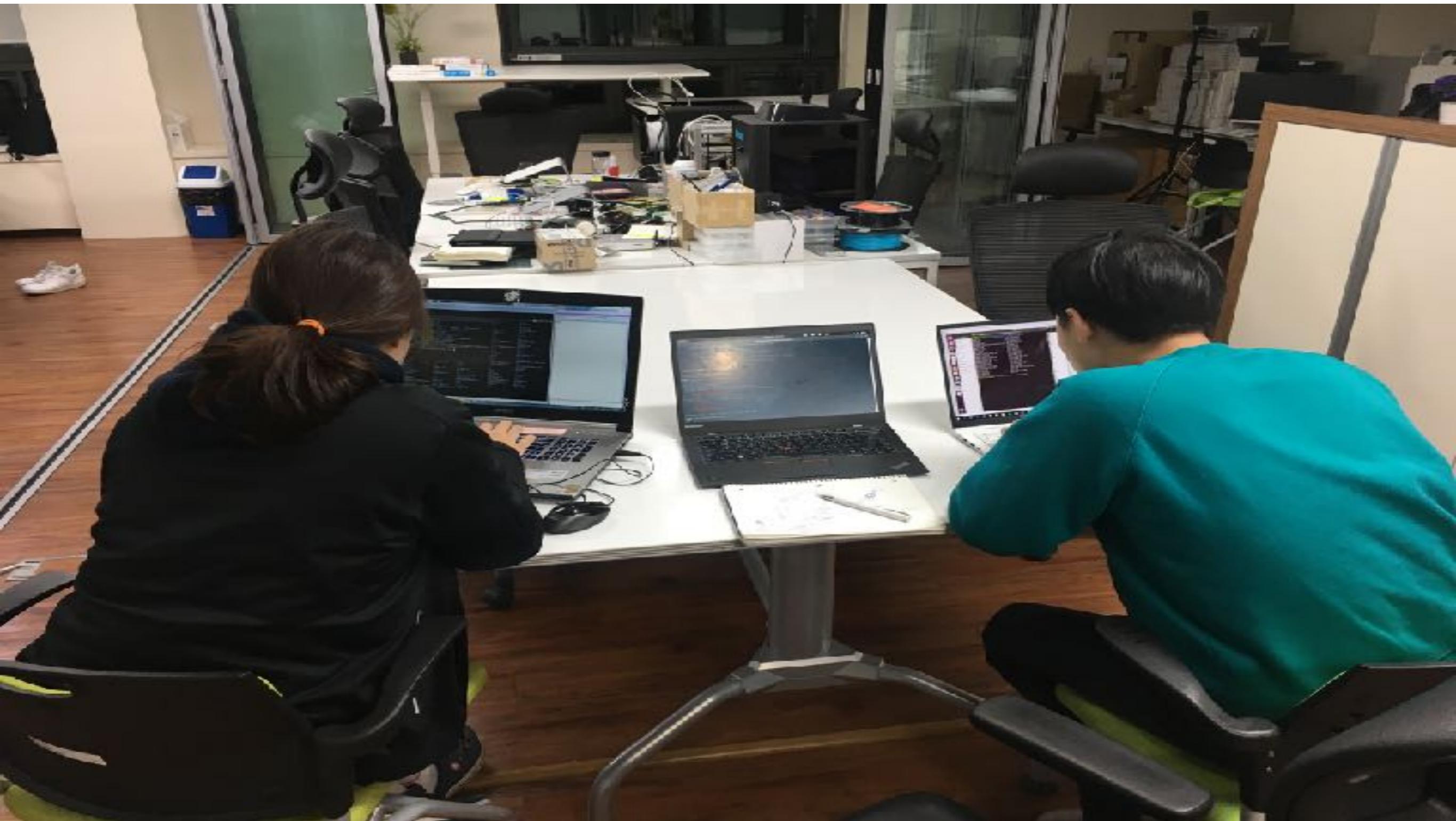


# 메인테이너와 만남

- 김남형 개발자(namhyung)
  - LG전자 오픈소스 팀
  - utrace 메인테이너
  - Linux Kernel; perf 공식 리뷰어



# KOSSLAB에서 밤샘 작업!



# 성과

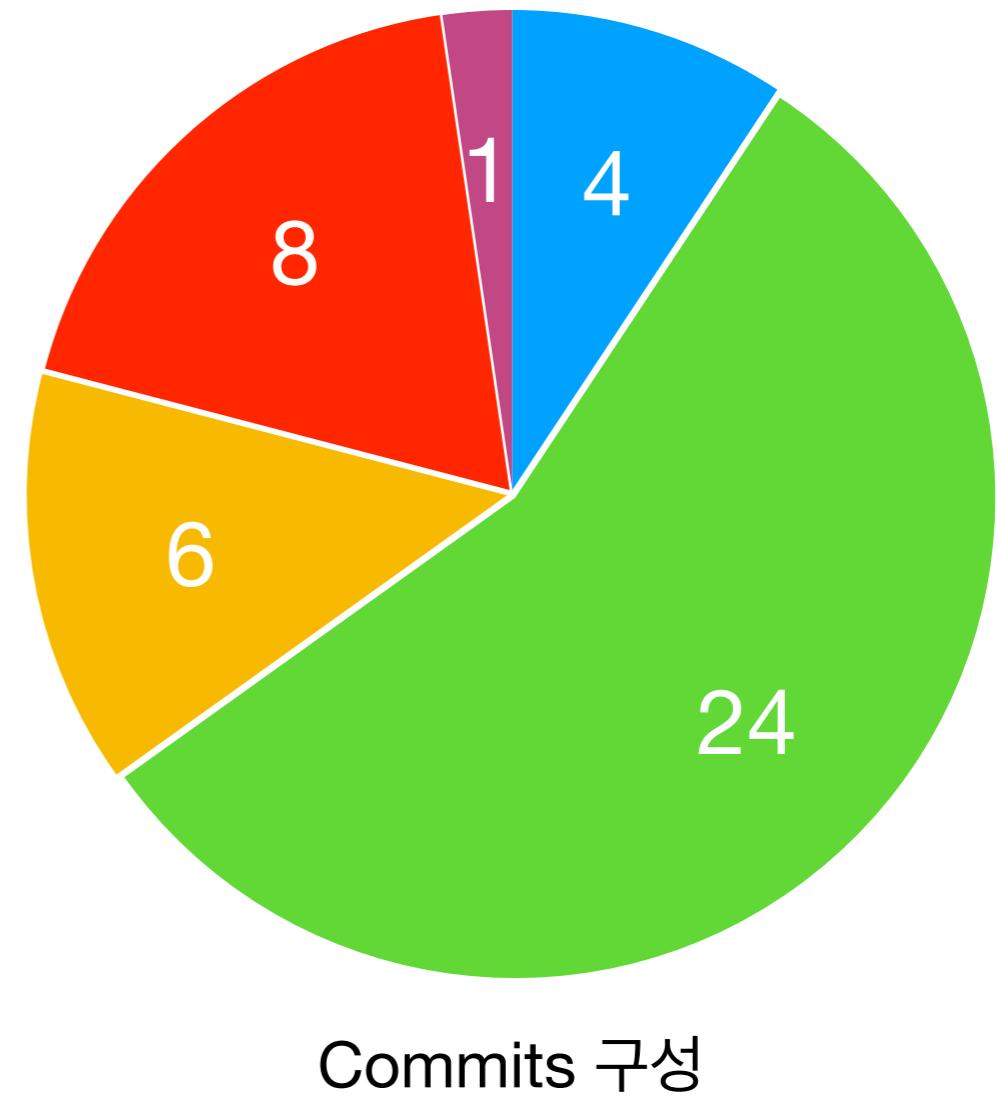
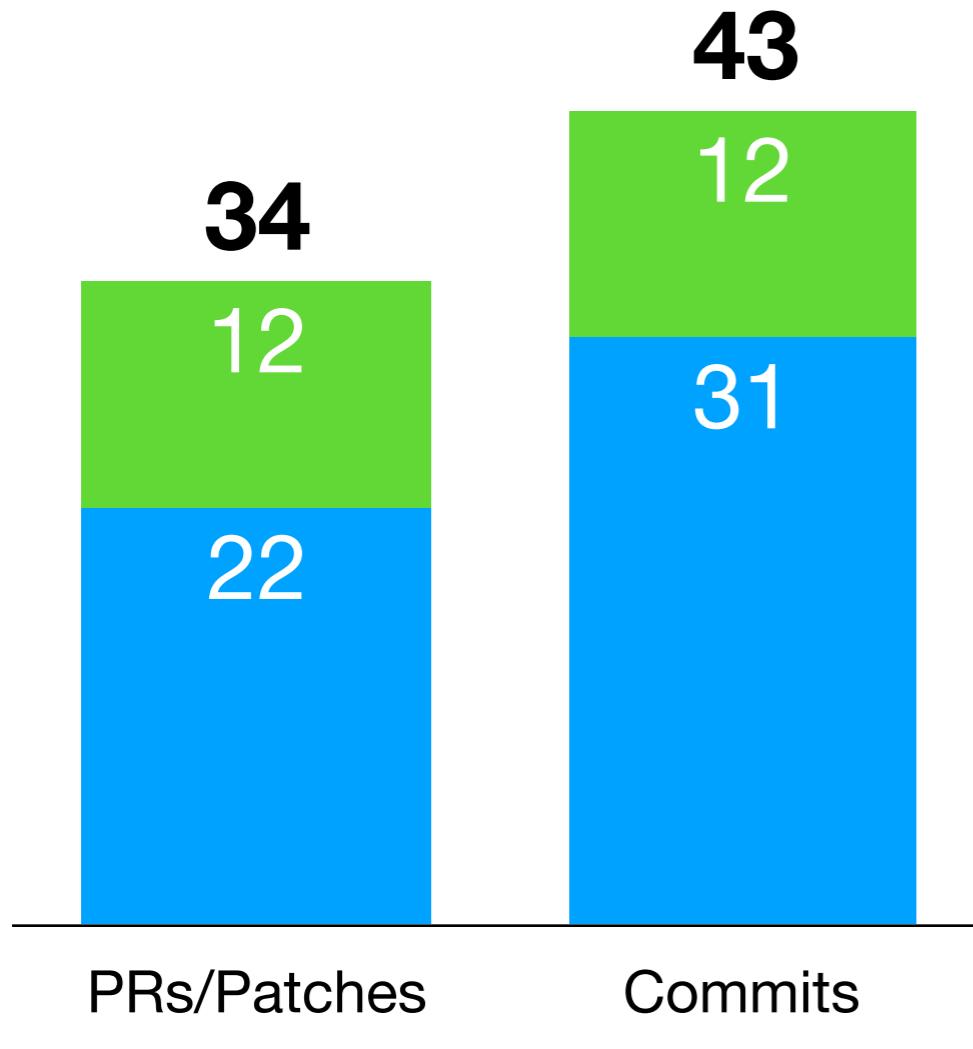
# PRs/Patches & Commits

uftrace

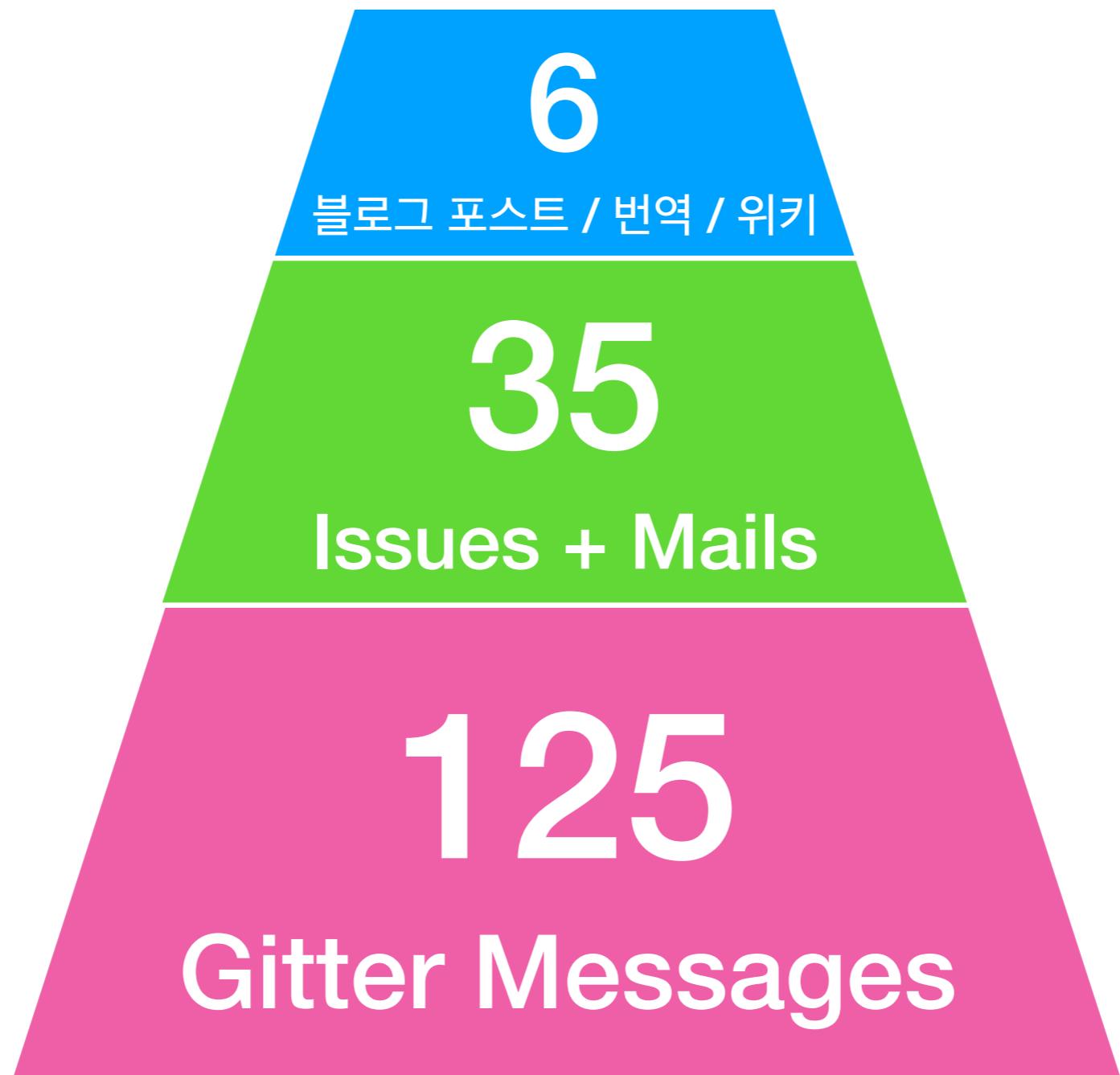
perf

리팩토링  
기능추가  
스페셜커밋

문서  
테스트



# Discussion



Feature1:  
32 비트 아키텍처(i386) 지원  
*(A commit makes a double scaled  
uftrace)*

# 기능추가; 32-bit 아키텍처 지원



박한범

Contributor

- 기존 uftrace 64-bit 아키텍처만 지원
  - X86\_64, ARM(v6~), AArch64
- i386 지원으로 32-bit 아키텍처 머신에서 활용 가능
  - 8086 아키텍처 임베디드 환경으로 확장 가능

# 기능추가; 32-bit 아키텍처 지원

Changes from 1 commit ~ Jump to... ~ +1 -1 Review changes

Unified Split

trigger: do not use memcpy() for copying trigger

```
if we use memcpy to copy the &iter->trigger,  
depending on the size of the uftrace_trigger +tr  
we will ride optimization path to use SSE inside memcpy.  
it will change status of SIMD register like mmx.  
therefore we lose original value of register.  
so, we decide not to use memcpy.
```

Signed-off-by: HanbumPark <kesell@gmail.com>

ParkHanbum committed 2 days ago commit a9420bfff8c2884a4fbb376530578718c6h88dd68

2 utils/filter.c

```
diff --git a/utils/filter.c b/utils/filter.c  
@@ -112,7 +112,7 @@ struct uftrace_filter *uftrace_match_filter(uint64_t ip, struct rb_root *root,  
112         iter = rb_entry(parent, struct uftrace_filter, node); 112         iter = rb_entry(parent, struct uftrace_filter, node);  
113  
114         if (match_ip(iter, ip)) { 113  
115 -             memcpy(tr, &iter->trigger, sizeof(*tr)); 114 +             *tr = iter->trigger;  
116     }  
117 }
```

- uftrace 구현상 memcpy(3) 내부 레지스터를 바꾸어 원하는 결과가 나오지 않음
  - 이에 대한 Issue가 없어 Troubleshooting이 오래걸림
  - 이를 Issue화 하여, memcpy(3) 전후에 레지스터를 복구하는 로직 추가

# 스페셜 커밋!; gcc 패치

- gcc i386 컴파일 시  
5bytes가 아닌 4bytes  
nop instruction을 만듬  
(bug 발견)
- inline asm 수정
- <https://gcc.gnu.org/ml/gcc-patches/2017-11/msg01259.html>

This is the mail archive of the [gcc-patches@gcc.gnu.org](mailto:gcc-patches@gcc.gnu.org) mailing list for the [GCC project](#).

Index Nav: [\[Date Index\]](#) [\[Subject Index\]](#) [\[Author Index\]](#) [\[Thread Index\]](#)  
Message Nav: [\[Date Prev\]](#) [\[Date Next\]](#) [\[Thread Prev\]](#) [\[Thread Next\]](#)  
Other format: [\[Raw text\]](#)

**[PATCH] fix -mnop-mcount generate 5byte nop in 32bit.**

*From: 박한범 <kese111 at gmail dot com>*  
*To: gcc-patches at gcc dot gnu dot org*  
*Date: Thu, 16 Nov 2017 06:24:15 +0900*  
*Subject: [PATCH] fix -mnop-mcount generate 5byte nop in 32bit.*  
*Authentication-results: sourceware.org; auth=none*

"-mnop-mcount" needs to make 5byte size "nop" instruction.  
however recently gcc make only 4byte "nop" in 32bit.  
I have test in gcc 5.4, 7.2.

=====  
bug result  
=====

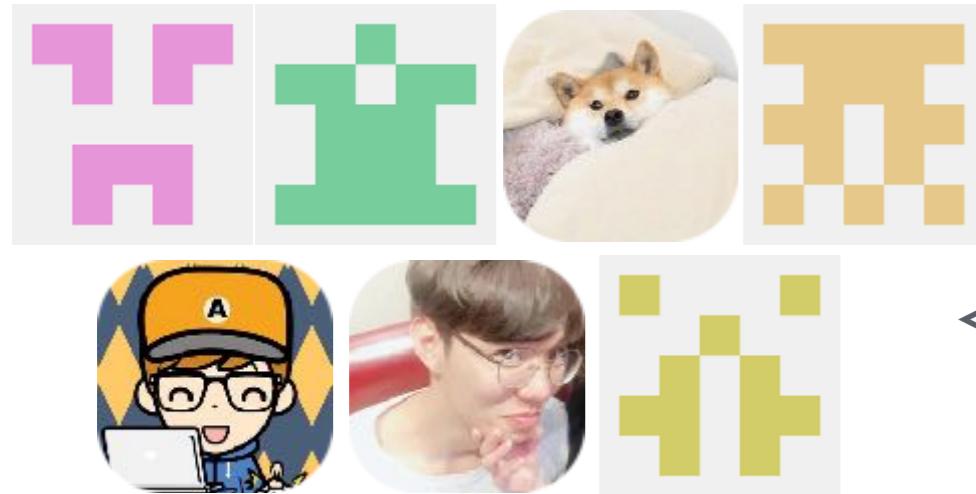
```
080485c5 <main>:  
00485c5: 0f 1f 04 00        nopl  (%eax,%eax,1)  
00485c9: 8d 4c 24 04        lea   0x4(%esp),%ecx  
00485cd: 83 e4 f0          and   $0xffffffff,%esp
```

=====  
fixed result  
=====

```
08048590 <main>:  
0048598: 0f 1f 44 00 01      nopl  0x1(%eax,%eax,1)  
004859d: 8d 4c 24 04        lea   0x4(%esp),%ecx  
00485a1: 83 e4 f0          and   $0xffffffff,%esp
```

**Feature2:**  
문서화 컨트리뷰션,  
컨트리뷰션을 위한 문서화  
*(Contribute for documentation,  
document for contribution)*

# 문서화 컨트리뷰션; uftrace

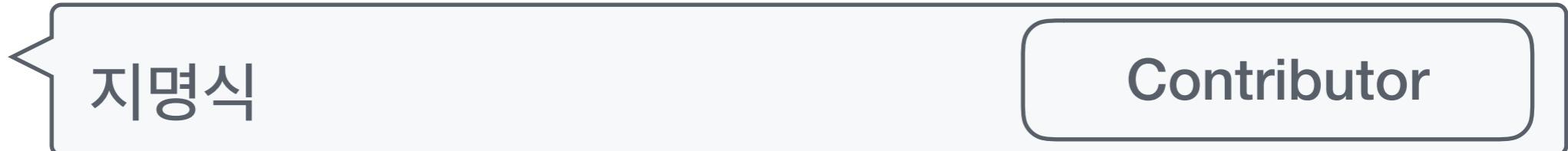
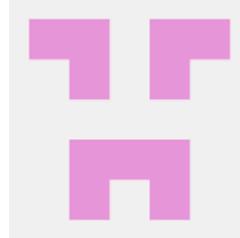


지명식, 장소진, 박한범,  
홍상원, 홍한석, 김유정,  
김태성

Contributor

- perf/uftrace 모두 명령 줄 도구이기 때문에 매우 중요!
- 새로 생긴 기능 중 사용법/예제가 빠진 것 추가
- 오류가 나는 예제 수정
- help 메세지 수정으로 사용자가 더 직관적으로 사용법 이해 가능

# README.md 예제의 오류 수정



- README.md 예제 오류 발견

Taeung Song (@Taeung)  
I think the executable 'fibonacci' doesn't mean exactly 'xcto/fibonacci'  
But I think you can use it,  
So we should change it with [@namhyung](#), [@hanggeekin](#)  
fibonacci > tests/fibonacci  
Because README and T1 file is like below

```
0: 00000000 <main>: 00000000 <main>
1: 00000000 <main>: 00000000 <main>
2: 00000000 <main>: 00000000 <main>
3: 00000000 <main>: 00000000 <main>
```

- Gitter를 통해 문의
- 예제 오류 수정 PR

databarn (@databarn)  
I was posting wrong case, here it

```
0x00000000 <main>: 00000000 <main>
0x00000001 <main>: 00000000 <main>
0x00000002 <main>: 00000000 <main>
0x00000003 <main>: 00000000 <main>
0x00000004 <main>: 00000000 <main>
0x00000005 <main>: 00000000 <main>
0x00000006 <main>: 00000000 <main>
0x00000007 <main>: 00000000 <main>
```

Namhyung Kim (@namhyung)  
@databarn just changing a case to pass the test is meaningless. In real world, 32 bit program will need to deal with 64-bit arguments.  
But I think the first implementation doesn't need to be perfect. Let's make the simple coursework and improve it later.  
[@JIMwang51](#), [@Young](#) I'm ok with the changes tests/fibonacci.

JIMwang51 (@JIMwang51)  
1. my test sample again. Then it work well. Thank you!  
2. I want to change like below. Please check.

```
git diff
diff --git a/main.c b/main.c
index 1111111..2222222 100644
--- a/main.c
+++ b/main.c
@@ -1 +1 @@
 #include <cs50.h>
```

# 옵션에 대한 예제 추가



장소진

Contributor

- 문서상 -pg 옵션 생략 수정
- script 예제 오타 수정
- -F 옵션 수정  
(기존에 -F 로도 backtrace가 가능했으나, 수정됨)
- uftrace-live, uftrace-replay의 time\_unit과 –color옵션 가능한 값 추가 & uftrace -f none 예제 추가



JangSoJin @JangSoJin

11월 09 11:14

Hi! 😊

I have one question & two suggestions about uftrace. (document)

one question is a result of `$ uftrace replay -F 'b@backtrace,depth=2'`. In Trigger section example of uftrace-replay.md (line 143~150), the result reads as follows.

```
$ uftrace record ./abc
$ uftrace replay -F 'b@backtrace,depth=2'
# DURATION      TID      FUNCTION
backtrace [ 1234] | /* [ 0] main */
backtrace [ 1234] | /* [ 1] a */
[ 1234] | b (
3.880 ns [ 1234] |   c();
5.475 ns [ 1234] | } /* b */
```

But when i executed it, my result was like below.

```
$ uftrace record ./abc
$ uftrace replay -F 'b@backtrace,depth=2'
# DURATION      TID      FUNCTION
[ 2243] | b() {
0.142 ns [ 2243] |   c();
0.745 ns [ 2243] | } /* b */
```

So i think it should be replaced by `$ uftrace replay -T 'b@backtrace,depth=2'` command, because of the result.

```
$ uftrace record ./abc
$ uftrace replay -T 'b@backtrace,depth=2'
# DURATION      TID      FUNCTION
0.789 ns [ 2243] | monostartup();
```

# 부족한 recv 사용법 문서에 대한 Issue화



장소진

Contributor

- uftrace recv; 문서상에 예제가 잘 나와있지 않아, 사용법에 대한 질문.
- 목적은 서로 다른 두 컴퓨터 사이의 data를 전달받아 tracing 하기 위함.
- 특히, -H는 uftrace record 부분에서 사용해야함.

JangSoJin commented 3 days ago

Hi, I have one question. :)  
I want to execute `uftrace recv`.  
When I wrote it, it didn't work. (nothing..)  
So... Is this command right? or wrong?

```
$ gcc -pg -o abc abc.c
```

(One terminal)

```
$ uftrace recv -H ubuntu-VirtualBox --run-cmd="uftrace replay"
```

(the other terminal)

```
$ uftrace record abc -H ubuntu-VirtualBox
```

If it is a wrong command, please let me know the correct example..

namhyung commented 3 days ago

The `recv` command aims to be run on a different machine. For example, it might have no enough space to save `uftrace` data on a small embedded device. Then it can send the data through network to a development (host) machine running `uftrace replay`. The `-H` option is for `uftrace record` to specify the name of the host.

honggyukim added the `question` label 3 days ago

# 명령어(recv, info) 사용법 추가



박한범

Contributor

- **uftrace recv**  
사용법 추가
- **uftrace info –symbol**  
사용법 추가

```
diff --git a/include/uapi/linux/uio.h b/include/uapi/linux/uio.h
index 0000000..1000000 100644
--- a/include/uapi/linux/uio.h
+++ b/include/uapi/linux/uio.h
@@ -1,4 +1,4 @@
 #ifndef _LINUX_UIO_H
-#define _LINUX_UIO_H
+/* This header is included by uio.h */
 #include <linux/types.h>
 #include <linux/uio.h>
```

across #include <linux/uio.h> use long long type for its first argument.  
In call to system, first argument is two block of static, so want to use argument.

```
diff --git a/tools/perf/parse-events.c b/tools/perf/parse-events.c
index 0000000..1000000 100644
--- a/tools/perf/parse-events.c
+++ b/tools/perf/parse-events.c
@@ -1,4 +1,4 @@
 #include "util.h"
-#include "util-sysfs.h"
+/* This header is included by util.h */
 #include <sys/types.h>
 #include <sys/sysinfo.h>
 #include <sys/conf.h>
```

```
diff --git a/tools/perf/parse-events.c b/tools/perf/parse-events.c
index 0000000..1000000 100644
--- a/tools/perf/parse-events.c
+++ b/tools/perf/parse-events.c
@@ -1,4 +1,4 @@
 #include "util.h"
-#include "util-sysfs.h"
+/* This header is included by util.h */
 #include <sys/types.h>
 #include <sys/sysinfo.h>
 #include <sys/conf.h>
```

# 기존 이슈 해결 및 업데이트



홍상원

Contributor

- 기존 Issue(#187) 해결
  - C++ 이름 규약을 따름
  - CONTRIBUTING.md에 변경된 URL 주고 PR

Naming convention of the C++ language standard. #229

Merged namhyung merged 1 commit into [namhyung:master](#) from [qpkzk:dev](#) 4 days ago

Conversation 9 · 0 Commits 1 · 1 File changed 15

qpkzk commented 5 days ago

I solved [#187](#) a little bit.  
I changed to naming convention of the C++ language standard in utrace.h file and utils/utils.h file.

namhyung commented 5 days ago

Owner +  
Thanks, but could you please do it for the all header files in a single commit?

qpkzk commented 5 days ago • edited

OK! I made one commit.  
By the way, do you want to change all header files in the utrace project like [#187](#).  
For example, replacing `_UTRACE_ASN_H_` with `UTRACE_ASN_H` in `asn.h`,  
replacing `_LINUX_LIST_H_` with `LINUX_LIST_H` in `list.h` and so on.

namhyung commented 5 days ago

Owner +  
Yes, please!

qpkzk commented 4 days ago

Contributor -  
I edited all header files in utrace project.

# 코드 규약과 관련한 이슈 제기



홍상원

Contributor

- 헤더 파일 중 #endif 옆에 대응하는 매크로 식별자를 주석으로 작성한 경우도 있는 반면 아닌 경우도 있어서 통일성을 위해 주석을 달려있지 않는 헤더 파일에 주석을다는 것이 어떤지 이슈를 제기함 그러나 받아들여지지 않음.

## Irregular comments next to preprocessors #236

Closed qpakzk opened this issue 4 days ago · 2 comments



qpakzk commented 4 days ago · edited

Contributor +

I discovered a few irregular comments next to preprocessors among header files.

First, almost all #endifs have [comments acknowledging what is matching macro identifiers](#), but a few #endifs don't have [them](#).

Second, some #elses have comments and others have [no comments](#).

And some #elses which own comments is [having ! in front of macro identifiers](#), but (the others) (<https://github.com/namhyung/uftrace/blob/master/utils/script-python.h#L20>) not.

Would you(@namhyung) mind adding comments and unifying them?



namhyung commented 4 days ago

Owner +

Actually I don't care much about these irregularities especially #else is placed closed to #if. Also codes in utils/list.h, utils/rbtree.h and libtraceevent/\* are copied from the Linux kernel source and I don't want to touch them unless there's a major issue.



qpakzk commented 4 days ago

Contributor +

OK! Thank you.



namhyung added the [wontfix](#) label 3 days ago



namhyung closed this 3 days ago

# 기본 값(default) 표기 에러나는 예제 수정



홍한석

Contributor

- `uftrace -help` 시 나오는 사용법에 기본값 추가(default)
- 사용자가 직관적으로 옵션 값을 ','로 구분하는걸 알 수 있음
- 오류나는 python script 예제 수정

The screenshot shows a GitHub pull request interface. At the top, it says "Merged" and "namhyung merged 2 commits into namhyung:master from flavono123:doc/help-defaults a day ago". Below this are tabs for "Conversation 1", "Commits 2", and "Files changed 1".  
  
The main area displays a comment by "flavono123" from 2 days ago:

Like the `--column-offset=DEPTH` line of current `uftrace --help` results, add default values of options, `-f [--output-fields=FIELD]` and `--diff-policy`, respectively.  
Following the written in man `uftrace-replay`, parenthesizing both default values with quote(')'s.  
I think the users may infer that the `FIELD` and `POLICY` can be separated by comma(,) from this.  
Now we can see the followed results of `uftrace --help`:

```
$ uftrace --help
Usage: uftrace [OPTION...]
      {record|replay|live|report|info|dump|recv|graph|script}
      [-p PROGRAM...]
uftrace -- function (graph) tracer for userspace

...
      --column-offset=DEPTH  Offset of each column (default: 8)
...
      --diff-policy=POLICY  Control diff report policy (default:
                            'abs,compact,no-percent')
...
      -f, --output-fields=FIELD Show FIELDs in the replay output (default:
                            'duration,tid')
...
```

Below the code block, there's another comment by "flavono123" from 2 days ago: "added some commits 2 days ago". It lists two commits:

- doc: Add the default value of "-f" option ... 3cbd431
- doc: Add the default of "--diff-policy" option ... 268b056

At the bottom, a checkmark icon indicates "namhyung approved these changes a day ago". A "View changes" button is also present.

# man page 예제 추가



김유정

Contributor

- **uftrace-replay.md의 FIELD 부분 옵션의 추가된 기본 기능에 대한 설명과 예시 추가**

The screenshot shows a GitHub pull request review thread. The first comment is from yujeongKim, who adds an example for the `uftrace-replay` command. The second comment is from namhyung, who suggests appending the example to the `FIELDS` section instead of the `TIME-RANGE` section because the `-r` option affects the output field. The third comment is from yujeongKim, thanking namhyung for the review and indicating they have modified the code.

```
graph TD; yujeongKim[ujeongKim] --> C1["Hi! @namhyung  
I add example for uftrace-replay.md. This mentioned at fd137a56b0  
What do you think about it? Thank you!"]; namhyung((namhyung)) --> C2["I think it'd be better to append this example to the FIELDS section as using -r option affects the output field too."]; yujeongKim --> C3["Thank you for reviewing! :) I modified!"]
```

Comments:

- yujeongKim commented 3 days ago:  
Hi! @namhyung  
I add example for `uftrace-replay.md`. This mentioned at [fd137a56b0](#)  
What do you think about it? Thank you!
- namhyung reviewed a day ago:  
I think it'd be better to append this example to the `FIELDS` section as using `-r` option affects the output field too.
- yujeongKim commented 6 hours ago:  
Thank you for reviewing! :) I modified!

# -f 옵션 예제 추가



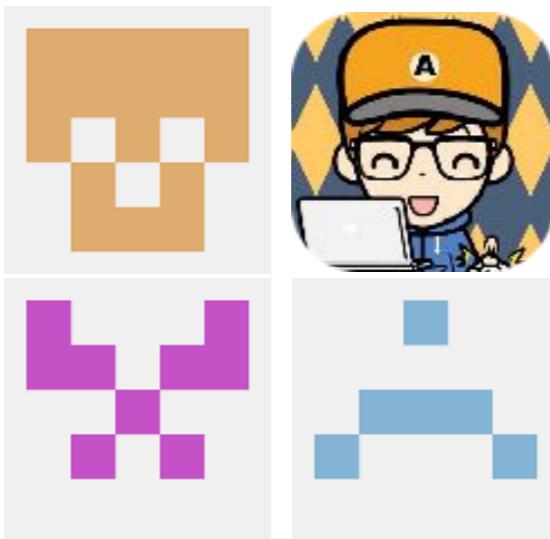
김태성

Contributor

- **uftrace replay -f 옵션 사용 예시 추가**

The screenshot shows a GitHub pull request page for a repository. The title of the pull request is "doc: Add -f option usage examples #253". The status is "Open" and it has 1 commit. The commit message is "Add -f option usage examples". The commit details mention that without the -f option, the result shows default fields, and it will show a chosen field by using -F option and add a field by using '-A' prefix. The commit was made by "jtd8274" 8 hours ago and has a green "Approved" status.

# 문서화 컨트리뷰션; perf



장시현, 홍상원, 양혜임  
임민채

Contributor

- 큰 프로젝트인 Linux Kernel에 비교적 낮은 진입 장벽
  - 첫 컨트리뷰션이 커널이라니!
- 또 다른 컨트리뷰션/컨트리뷰터를 위한 문서 작업([wiki](#))

# perf trace: tip 추가



홍상원

Contributor

- perf trace에 대한  
tip 추가

[PATCH] perf trace: Add tips about perf trace



Sangwon Hong <qpakzk@gmail.com>

Arnaldo, linux-kernel, Jiri, Namhyung, Taeung에게

Add two tips that summary perf trace and perf trace record to know intuitively.

Cc: Jiri Olsa <[jolsa@kernel.org](mailto:jolsa@kernel.org)>

Cc: Namhyung Kim <[namhyung@kernel.org](mailto:namhyung@kernel.org)>

Cc: Taeung Song <[taeungi@osslab.kr](mailto:taeungi@osslab.kr)>

Signed-off-by: Sangwon Hong <[qpakzk@gmail.com](mailto:qpakzk@gmail.com)>

---

tools/perf/Documentation/tips.txt | 2 ++

1 file changed, 2 insertions(+)

diff --git a/tools/perf/Documentation/tips.txt b/tools/perf/Documentation/tips.txt

index db0ca30..a9cd3f4f 100644

--- a/tools/perf/Documentation/tips.txt

+++ b/tools/perf/Documentation/tips.txt

@@ -32,3 +32,5 @@ Order by the overhead of source file name and line number: perf report -s smcln

System-wide collection from all CPUs: perf record -a

Show current config key-value pairs: perf config --list

Show user configuration overrides: perf config --user --list

+Show system events including syscalls: perf trace

+To generate perf.data files including raw\_syscalls events, try: perf trace record

...

--  
2.7.4

# doc: tip 추가



양혜임

Contributor

- –print-line 옵션에 대한 tip 추가



Hyeim Yang <yanghyeimm@gmail.com>

Arnaldo, linux-kernel, Jiri, Namhyung, Taeung에게 ↗

Cc: Jiri Olsa <jolsa@kernel.org>

Cc: Namhyung Kim <namhyung@kernel.org>

Cc: Taeung Song <taeung@kusslab.ki>

---

tools/perf/Documentation/tips.txt | 1 +  
1 file changed, 1 insertion(-)

```
diff --git a/tools/perf/Documentation/tips.txt b/tools/perf/Documentation/tips.txt  
index d01ca30..b60616a 100644
```

```
--- a/tools/perf/Documentation/tips.txt
```

```
+++ b/tools/perf/Documentation/tips.txt
```

```
@@@ 32,3 +32,4 @@@ Order by the overhead of source file name and line number. perf report -s srokin
```

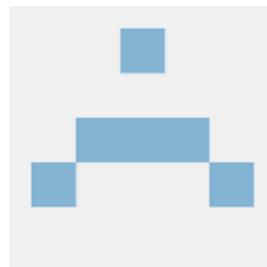
```
System-wide collection from all CPUs: perf record -a
```

```
Show current config key-value pairs: perf config --list
```

```
Show user configuration overrides: perf config --user --list
```

```
Print matching source lines (may be slow): perf annotate --print-line <symbol>
```

# perf probe: tip 추가



임민채

Contributor

- perf probe에 대한  
tip 추가

[PATCH] perf probe: Add a tip about define a tracepoint event □ 민은관시함 x

minchae <mr.comkorea@gmail.com>  
Amaldo, Ilinux-kernel, JIrl, Namhyung, Taeung, minchae에게 □

perf probe command defines dynamic tracepoint events, by symbol and registers  
without debuginfo, or by C expressions with debuginfo.

— tools/perf/Documentation/tips.txt | 2 +-  
1 file changed, 1 insertion(+), 1 deletion(-)

```
diff -git a/tools/perf/Documentation/tips.txt b/tools/perf/Documentation/tips.txt  
index da585ac..8487170 100644  
--- a/tools/perf/Documentation/tips.txt  
+++ b/tools/perf/Documentation/tips.txt  
@@ -32,5 +32,5 @@ Order by the overhead of source file name and line number: perf report -s srcfile  
System-wide collection from all CPUs: perf record -a  
Show current config key-value pairs: perf config --list  
Show user configuration overrides: perf config --user --list  
-Define a dynamic tracepoint event: perf probe -a  
+Define a dynamic tracepoint event: perf probe --add
```

# man page 누락된 내용 추가



장시현

Contributor

- perf top
- perf lock
- perf inject
- perf trace
- perf time chart
- perf sched
- perf evilest
- perf builds

index : kernel/git/acme/linux.git

perf, networking and other medium stuff

about summary refs log tree commit diff stats

log msg search

| Author                   | Files | Lines   |
|--------------------------|-------|---------|
| Jim Olisa                | 1     | -0/+6   |
| Jim Olisa                | 1     | -0/+8   |
| Jim Olisa                | 3     | -0/+7   |
| Jim Olisa                | 2     | -10/+9  |
| Jim Olisa                | 1     | -0/+0   |
| Jim Olisa                | 1     | -1/+2   |
| Jim Olisa                | 1     | -12/+8  |
| Jim Olisa                | 1     | -0/+7   |
| Jim Olisa                | 5     | -88/+27 |
| Jim Olisa                | 4     | -64/+8  |
| Jim Olisa                | 1     | -4/+2   |
| Jim Olisa                | 1     | -1/+1   |
| Sihyeon Jang             | 1     | -0/+3   |
| Sihyeon Jang             | 1     | -0/+4   |
| Sihyeon Jang             | 1     | -0/+4   |
| Sihyeon Jang             | 1     | -1/+1   |
| Sihyeon Jang             | 1     | -6/+10  |
| Sihyeon Jang             | 1     | -0/+4   |
| Jim Olisa                | 1     | -1/+4   |
| Sihyeon Jang             | 1     | -0/+4   |
| Kim Phillips             | 1     | -0/+1   |
| Kim Phillips             | 1     | -4/+4   |
| Arnaldo Carvalho de Melo | 1     | -3/+5   |
| Arnaldo Carvalho de Melo | 1     | -12/+14 |
| Arnaldo Carvalho de Melo | 1     | -0/+8   |
| Arnaldo Carvalho de Melo | 2     | -0/+14  |
| Thomas Richter           | 1     | -2/+5   |
| Thomas Richter           | 1     | -2/+4   |
| Ravi Bangora             | 1     | -0/+9   |
| Namryung Kim             | 1     | -2/+2   |
| Arnaldo Carvalho de Melo | 1     | -0/+3   |
| Wang Nan                 | 1     | -1/+1   |
| Sihyeon Jang             | 1     | -1/+1   |
| Sihyeon Jang             | 1     | -0/+6   |

# 컨트리뷰션을 위한 문서화;

## Wiki: First step being a contributor of perf

- perf(Linux 프로젝트) 컨트리뷰션을 위한 명령어 단위의 상세한 설명
- 실제로 이후 팀원들이 컨트리뷰션 시 참고
- <https://github.com/sihyeonn/linux-perf/wiki/First-step-to-being-a-contributor-of-perf>

First step to being a contributor of perf  
sihyeonn edited this page 8 days ago - 3 revisions

Welcome to the linux-perf wiki!

### Environment

- OS : Linux mint 18.2 sonya Mate
- Kernel : 4.10.0-37-generic

### Process to contribute

```
$ git clone https://github.com/sihyeonn/linux-perf.git
```

2. Add remote repository.

```
$ cd linux-perf  
$ git remote add some git://git.kernel.org/pub/scm/linux/kernel/git/acme/linux.git
```

3. Get the kernel sources and install perf.

```
$ git fetch some perf/core  
$ cd tools/perf  
$ sudo apt-get install -y libdw-dev libelf-dev libnewt-dev libunwind8-dev elfutils  
$ sudo make  
$ sudo make install
```

Or if you want to trace perf with [utrace](#), you can use '-pg' options.

```
$ git fetch some perf/core  
$ cd tools/perf  
$ sudo apt-get install -y libdw-dev libelf-dev libnewt-dev libunwind8-dev elfutils  
$ sudo make EXTRA_CFLAGS='-pg'  
$ sudo make install EXTRA_CFLAGS='-pg'
```

Set Git email

# 컨트리뷰션을 위한 문서화; 블로그 포스트

Perf\_1

06/2017 | min read

'Perf' tutorial"



언제나 시작

REVIEW

개발에 경향나와 기록

home / archive / categories / tips

✉ f o g

여기 리눅스 커널 중 System의 Performance를 측정하는 Tool인 'Perf'에 대해 살펴보자.

또한, 이 글이 유용한 대상은 시스템 프로그래밍, 보안, tracing에 관심이 있는 학생 혹은 초보 개발자일 것이다. 나의 일은 지식으로는 거기까지 간접적으로 나의 도움을 줄 수 있는 범위라고 생각되기 때문이다. ;)

사전에 준비해야 할 것은 'git', 'kraae', 'l' (컴파일러) 정도가 될 것이다. 이게 대해서는 다른 나의 빙고기를 참조하라.

수기적으로 contribution이 주가 될 것이기 때문에 git과 컴퓨터 시스템에 대한 전반적인 이해를 통한 한나연풀사의 성장과 함께할 수 있을 것이다.

그리니 여기 글은 git과 컴퓨터 시스템에 대해 다룬 금부히도록 하자(필자의 경우도 별도로 공부할 것이다).

다소 모호할 수 있는 '전반적인 이해'는 '어느정도 숙달된'을 의미한다. 이를 예로 들면 release-ci를 이용하여 고려 커밋을 수정할 수 있는 단계라고 이해하면 된다. 컴퓨터 시스템의 경우에는 컴퓨터의 내부 구조와 함께 OS 혹은 kernel이 작용하는 과정에 대해 이해하고 있는 것을 의미할 것이다.

## perf 를 1도 모르지만 커밋하고 싶어 공부하기

제목 그대로입니다. 컨트리뷰션에 참가했지만 시험 등 이런저런 일정과 핑계와 게으름에 차이다 보니 지금까지 커밋은커녕 그를 위한 제대로 된 공부도 못했습니다. 그래서 공동 구글 드라이브에 있는 참고자료와 빌드한 [perf](#) 를 바탕으로 공부하며 정리하려 합니다.

### perf 의 개발 저장소/브랜치 클론하고 빌드하기

알림: 사소한 용어 정리; 둘 다 "설치"라고 해석될 여지가 있어 "클론"과 "빌드"로 나누었습니다. 클론?:

`git clone` 입니다. 개발(커밋)을 위해 소스파일이 있는 저장소를 지역 저장소로 복사합니다.

`perf` (리눅스 커널 프로젝트)는 git으로 관리되기 때문입니다.

빌드?: `make` 입니다. 받은 또는 개발(수정)이 된 소스를 컴파일해서 바이너리로 만듭니다.

`perf` 를 1도 모르지만 일단 받아 놓고 봅시다.

### 개발 저장소( tip ) 클론하기

λ `cd <_적당한_개발_디렉토리_>`

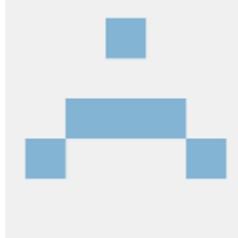
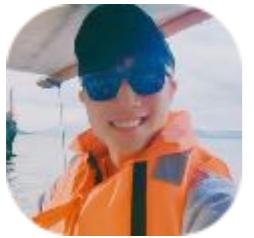
λ `git clone git://git.kernel.org/pub/scm/linux/kernel/git/tip/tip.git -b perf/core`

# Feature3:

## 버그 발견 및 테스트

*(Why always me?...)*

# 버그 발견 및 테스트



고명진, 박한범, 임민채,  
이주선

Contributor

- 다양한 환경(리눅스 배포판, 가상머신, ...)에서 진행하다 보니 버그 발견
- 버그 reports 및 테스트에 대한 Issues 작성
- 프로그램 내 테스트 기능 활용, 새로운 기능에 대한 테스트 PR

# pager와 VMWare 환경에서 버그 report(Issues)



이주선

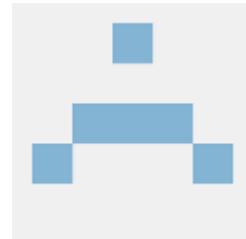
Contributor

- 결과 출력하는 pager와 interaction이 잘 안되는 문제 케이스 제공
- Kernel trace 옵션이 특정환경(VMWare)에서 에러남을 Issue화
- 바이너리 취약점분석이나 시큐어코딩에 도움되는 방향으로 이슈를 제기함

The screenshot shows a GitHub issue thread with four comments:

- namhyung commented 26 days ago**  
Hello, utrace ("live" command in this case) uses a pager (default: less(1)) to help users reading the output (from your program and utrace). But it hinders interaction via stdin unfortunately. You can use --no-pager option to disable the pager - I don't know how to handle it automatically. Note that utrace record command disables the pager by default.
- elfring commented 26 days ago**  
You can use --no-pager option to disable automatically.  
User - Process interaction problem on [command-line option] - Issue #197 - handle it  
Could you get further inspiration from the content management software "Git"?
- namhyung commented 26 days ago**  
@elfring well, my concern was how to detect from utrace for child to use stdin. Do you know how git deals with it (or is there a case for git worry about it)?
- elfring commented 26 days ago**  
Did you notice that Git tools change their behaviour depending on output file redirection?

# 멀티 쓰레드 관련 버그 report(Issues/PR)



임민채

Contributor

- 멀티-쓰레드 상황에서 `pthread_exit()`을 활용하여 thread 내부를 탈출할 때 Segmentation Fault가 발생하는 bug 발견
- 그에 대한 간략한 코드 작성 및 Issues 등록
- 위 문제가 있는 Testcase에 대한 PR



Namhyung Kim @namhyung

우선 `pthread_exit()` 호출 시 `utrace` 내부 자료 구조를 복구하면 `segfault`는 발생하지 않네요.

```
diff --git a/libcount/pithook.c b/libcount/pithook.c
index f2e586f..5cce849 100644
--- a/libcount/pithook.c
+++ b/libcount/pithook.c
@@ -374,10 +388,11 @@ static const char *flush_syms[] = {
     "fork", "vfork", "daemon", "exit",
     "longjmp", "siglongjmp", "__longjmp_unk",
     "exec!", "execp", "execle", "execv", "execve", "execvp", "execvpe",
+    "pthread_exit",
};

static const char *except_syms[] = {
-    "_Unwind_RaiseException",
+    "_Unwind_RaiseException", "pthread_exit",

static void add_special_func(struct pithook_data *pd, unsigned idx, unsigned flags)
```



minchae @minchae

혹시 그때도 `result`값을 제대로 출력하는지 확인해주실 수 있나요?

제가 확인한바로는 `utrace`로 돌리면 `double`형 인자를 출력할 경우 무조건 0을 출력하게 되는 오류가 있습니다

11월 15 13:30 ✓

11월 15 13:45

# 기존 tests에 대한 warning 수정 zsh-completion 기능 추가 중 버그 발견



고명진

Contributor

- 기존 tests를 위한 소스 (s-alloc.c, s-fork.c, s-malloc-fork.c, s-ucontext.c) 컴파일 시 warning 발견.

- 이를 수정한 PR - Commit message 수정과 서명을 rebase로 해결

- Zsh-completion(자동완성) 기능 추가 중, Bash-completion이 설치되지 않음에도 install 되었다는 조건적 버그 발견, Issue 작성

```
HansukHong @flavon123 Nov 15 14:02
Hi all, I wondered tracing a utrace by another utrace is possible!
$ UTRACE_UTRACE </path/to/bin> # <- It's my own program
utrace: /home/flavon123/git/utrace/cmd-record.c:146:check_binary
ERROR: Can't find 'mcOUNT' symbol in the 'utrace'.
It seems not to be compiled with -fPI or -fInstrument-functions flag
which generates traceable code. Please check your binary file.

It tells me need to compile the utrace with -fPI option, but I've installed/compiled the utrace with
$ sudo make UTRACE_CFLAGS=-fPI -fPI install # as utrace project root
And clearly, $ utrace </path/to/bin> is executable. Thanks for attentions 😊

MyungjinHo @j1197 Nov 15 15:22
In my case, I added the -sg option to the path in MAKEFILE like below.

# Note that the plain CFLAGS and LDFLAGS can be changed
# by config/makefile later but *_FLAGS can not.
#
UTRACE_CFLAGS = ${COMMON_CFLAGS} ${CFLAGS_68} ${CFLAGS_utrace} -sg
TRACEEVENT_CFLAGS = ${COMMON_CFLAGS} ${CFLAGS_39} ${CFLAGS_traceevent}
LIB_CFLAGS = ${COMMON_CFLAGS} ${CFLAGS_80} ${CFLAGS_LIB} -fPIC -fvisibility=hidden
...
but, I don't know what's different from your case. As a result, this is traceable.

HansukHong @flavon123 Nov 15 15:29
@j1197 Right!
Recompiled with the Makefile for you've mentioned, it works!
Thanks! 🎉
```

# 32-bit 아키텍처(i386)를 위한 testcase 추가



박한범

Contributor

- 새로 추가하신 기능인  
32-bit 아키텍처(i386)과  
관련된 testcase 추가



de4taoam @ParkHanbum

is test case 061 consider type casting? or not? Because long is not bigger than unsigned in 32bit.

therefore, I get test result like this:

```
16 ====== result ======
17 main() {
18     int_add(-1, 2) = 1;
19     int_sub(1, 2) = -1;
20     int_mul(3, 4) = 12;
21     int_div(4, -2) = 0;
22 } /* main */
23 ====== expected ======
24 main() {
25     int_add(-1, 2) = 1;
26     int_sub(1, 2) = -1;
27     int_mul(3, 4) = 12;
28     int_div(4, -2) = -2;
29 } /* main */
```

so, now I change test code

```
16 int int_div(unsigned a, long b)
17 {
18     return a / b;
19 }
```

to

```
16 int int_div(int a, int b)
17 {
18     return a / b;
19 }
```

Is it ok to change this way?



Namhyung Kim @namhyung

@ParkHanbum I don't know why it makes a difference in the return value.



de4taoam @ParkHanbum

@namhyung I don't ask why it does.

the reason is automatic type casting.

my question is "this test intest type casting"

"s intent type casting?"

and is it ok to change argument type to int?



Namhyung Kim @namhyung

@ParkHanbum hi, what "type casting" do you say? the intention was to verify utrace can trace arguments of various types.



de4taoam @ParkHanbum

is it ok to change argument type to int?

11월 01 01:08

11월 01 23:44

11월 02 10:06

11월 02 10:10

11월 02 10:13

## Feature4:

리팩토링; 소스 코드 읽기 및 분석

*(Any fool can write code that a computer  
can understand. Good programmers write  
code that humans can understand.)*

# 리팩토링



박성현, 홍상원

Contributor

- 소스 코드 읽기와 분석을 통한 리팩토링
  - 코드 분석 기술: 에디터 활용, 심볼 추적 (etags, cscope)
  - git blame, checkout, tag, reset, ...
- 기존 열린 Issue를 해결하며 소스 코드 분석

# perf; 누락된 구조체 필드 추가



박성현

Contributor

- 퍼포먼스 카운터 값을 읽어오는 시스템 콜을 찾던 중 perf\_event.h의 perf\_attr\_open 함수가 사용됨
- 이 함수에 인자로 넘겨지는 struct perf\_event\_attr에 새롭게 추가된 필드에 대해서는 복사하는 코드가 누락돼 있었음. 이 부분을 추가하는 패치 메일을 작성하기로 결정

author: Seonghyun Park <seonghyun0p@gmail.com> 2017-11-09 23:07:04 +0900  
committer: Arnaldo Carvalho de Melo <acme@redhat.com> 2017-11-13 09:40:04 -0300  
commit: 025541c64438e4cd4f4d982391a68417a59c74cf (patch)  
tree: 076630ae9c9778fb57b1b39359736b3eda5c94a  
parent: 99396cc1279eeba1f09a09d189adca551510183f (diff)  
download: linux-025541c64438e4cd4f4d982391a68417a59c74cf.tar.gz

perf tests: Add missing WRITE\_ASS for new fields of perf\_event\_attr

Include newly added fields 'mmap2', 'comm\_exec', 'use\_clockid', 'namespaces', 'write\_backward' and 'context\_switch' from perf\_event\_attr to store\_event().

Signed-off-by: Seonghyun Park <seonghyun0p@gmail.com>  
Cc: Jiri Olsa <jolsa@kernel.org>  
Cc: Namhyung Kim <namhyung@kernel.org>  
Cc: Seonghyun Park <seonghyun0p@gmail.com>  
Link: http://lkml.kernel.org/n/tip-7pgcv0h5fmc9cthk87q@git.kernel.org  
[ Fix log message to add 'write\_backward', fix the patch to add 'use\_clock\_id' ]  
Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>

## Diffstat

+rw-r--r-- tools/perf/tests/attr.c 6

1 files changed, 6 insertions, 0 deletions

```
diff --git a/tools/perf/tests/attr.c b/tools/perf/tests/attr.c
index 0e1367f..97f64ad 100644
--- a/tools/perf/tests/attr.c
+++ b/tools/perf/tests/attr.c
@@ -124,6 +124,12 @@ static int store_event(struct perf_event_attr *attr, pid_t pid, int cpu,
         MRITE_ASS(exclude_guest, "d");
         MRITE_ASS(exclude_callchain_kernel, "d");
         MRITE_ASS(exclude_callchain_user, "d");
+        MRITE_ASS(mmap2, "d");
+        MRITE_ASS(comm_exec, "d");
+        MRITE_ASS(context_switch, "d");
+        MRITE_ASS(write_backward, "d");
+        MRITE_ASS(namespaces, "d");
+        MRITE_ASS(use_clockid, "d");
         MRITE_ASS(wakeup_events, PRTu32);
         MRITE_ASS(bp_type, PRTu32);
         MRITE_ASS(config1, "llu");
```

# uftrace; 기존 Issue를 해결



홍상원

Contributor

- 이슈 분석

- fopen()의 반환값이 NULL이 되는 경우 에러 처리를 해달라는 이슈
- printf()를 pr\_out()으로 수정해달라는 이슈

Solve issue #194 . #224

Merged namhyung merged 2 commits into namhyung:master from qpakzk:err-check-v1.c 9 days ago

Conversation 3 Commits 2 File changed 2

The screenshot shows a GitHub pull request interface for issue #194. At the top, it says "Solve issue #194 . #224" and "Merged namhyung merged 2 commits into namhyung:master from qpakzk:err-check-v1.c 9 days ago". Below this are tabs for "Conversation 3", "Commits 2", and "File changed 2".  
The first commit by qpakzk is shown with a comment: "Hello @namhyung , I send this PR. I solved issue #194 . First, I added error handling for return values from fopen in session.c. Second, I changed functions from to pr\_out in symbol.c." It also includes a "Signed-off-by: Sangwon Hong qpakzk@gmail.com" and a thumbs-up icon.  
The second commit by qpakzk is shown with a comment: "qpakzk added some commits 9 days ago" followed by two commit details:

- symbol: Change From printf to pr\_out ... (commit hash: 88fe327)
- session: Add error handling for return values from fopen ... (commit hash: 693d346)

  
A review by namhyung follows, with a "View changes" button and a "Show outdated" button.  
A reply by qpakzk edits the previous comment: "I edited to TEST\_NE(fp, NULL) and changed indentation the way you had pointed out."  
A final review by namhyung is shown with the status "LGTM".  
At the bottom, it says "namhyung merged commit da5d9e6 into namhyung:master 9 days ago" with "View details" and "Revert" buttons.

Feature5:  
Git/Github을 사용한  
오픈소스 개발, 협업

*(Open Source development = Co-working  
= Professional Git/Github usage)*

# Git을 활용한 협업; master 아닌 브랜치에 PR하기



홍한석

Contributor

- 인자 출력 옵션에 enum, struct 형식 출력 기능을 추가하려 함
- 관련 Issue(#174)가 이미 있어 토의에 참여
  - 메인테이너가 이미 개발하고 계심!!
  - 해당 기능의 일부가 구현되어 있지만 master가 아닌 별도의 브랜치 → ... conflicts!

# Git을 활용한 협업; master 아닌 브랜치에 PR하기

- 메인테이너께 **rebase** 요청

- 당장은 아니고,  
작업이 끝난 후에 가능하다 말  
씀

- **Professional Git/Github**  
사용으로 해결

- cherry-pick, format-patch
- **Comparing changes**만들기  
(포크 된 저장소와 기존 저장소)

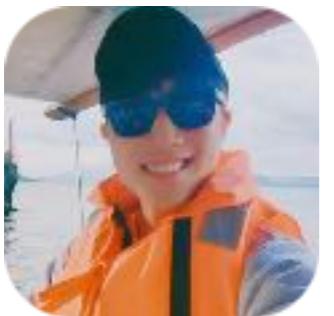
## Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

The screenshot shows a GitHub interface for comparing branches. At the top, there are dropdown menus for 'base fork: namhyung/utrace', 'base: master', and 'head fork: flavon0123/utrace', along with a 'compare' dropdown set to 'dwarf-argspec-func-ptr'. A red error message 'X Can't automatically merge. Don't worry, you can still create the pull request.' is displayed. Below this, a green button says 'Create pull request'. To the right, it says 'Discuss and review the changes in this comparison with others.' Below the button, it shows '19 commits', '23 files changed', and '0 commit comments'. The commit history shows three commits from 'namhyung + flavon0123' on Nov 16, 2017, and one commit from 'flavon0123'. To the right of the commits, there are three links: 'trigger: Add 'p' specifier for function ptr args ...', 'replay: Implement function pointer argspec ...', and 'dwarf: Support ARG\_FMT\_FUNC\_PTR arg format ...'.

느낀점

# Contribution의 가치를 알게되다



고명진

Contributor

uftrace/perf를 공부하는 과정에서 선배 개발자의 코드를 공부하며 그들이 얼마만큼의 정성을 들여 프로그램을 개발했고, 개발하는 중이라는 것을 느끼게 되었습니다. 이러한 경험은 제게 그들이 정성스레 빚은 작품에 기여하고 있다는 가치를 주었고, 그들과 함께 프로그램을 만들고 있다는 영광도 주었습니다.  
이번 계기를 통해 uftrace와 더불어 주로 사용하는 오픈 소스 프로그램에 기여할 용기를 얻게 되어 기쁩니다.

# Thank you Contributhon

## 2017



김유정

Contributor

고급개발자이신 멘토님으로부터 직접 실전적으로 Git을 사용해 볼 수 있어서 초보개발자로서 큰 배움의 경험이 되었습니다. 😊

# 오픈소스 개발자로서의 첫걸음



박성현

Contributor

멘토님께서 체계적으로 git 사용법 및 오픈소스 분석 방법에 대해 교육해주셔서, 처음 오픈소스를 접하는 입장에서도 공부하기 좋았습니다.  
학기 중이라 많은 시간을 투자하지 못한 게 아쉽고, 좀 더 실력을 쌓아서 활발하게 오픈 소스에 기여하고 싶다고 느꼈습니다.

# 이슈가 많을수록 좋은 오픈소스가 되다

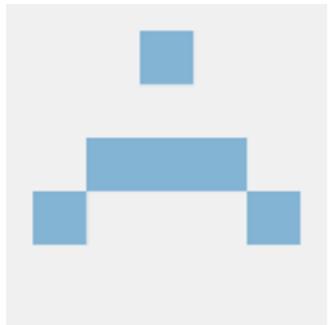


박한범

Contributor

memcpy(3)를 사용해 commit을 작성하는데 원치 않는 결과가 나와 버그인줄 알았습니다. 그러나 uftrace 내부 구현이 그리 되어 있었고, 이에 대한 이슈가 없어 파악하는데 힘들고 오래걸렸습니다.  
새로운 기여자의 참여로 더 좋은 오픈소스가 되기 위해 이슈를 많이 올리는 것이 좋다는 교훈을 얻었습니다.

# Thank you Contribution 2017



임민채

Contributor

저희학교 교수님이 컴퓨터 사이언티스트로 돈,명예를 얻는 방법은 3가지가 있다하셨는데  
첫번째가, 세계적인 논문 쓰기  
두번째가, 멘토님 같은 굵직한 오픈소스 개발자가 되기  
세번째 방법은 창업이라고 하셨습니다.  
특히 첫번째 방법을 강조하며 자기 밑 대학원 오라고 꼬셔서 그렇게 신빙성은 없지만  
이번 컨트리뷰톤을 참여하면서 두번째 방법인 오픈소스  
개발자가 되는 기틀을 닦은거 같아  
송태웅 멘토님과 김남영, 김홍규 개발자님께 감사드립니다.

# Thank you Contributhon

## 2017



장소진

Contributor

처음으로 참여하는 오픈소스 프로젝트라 떨리기도 하고,  
혹여나 민폐가 되지 않을까 걱정이 많이 되었는데 멘토님  
과 팀원분들의 도움덕분에 다양하게 `uftrace`를 사용해볼  
수 있었습니다. 😊 아직은 기술적으로 부족한 부분이 많  
아 `document` 위주의 간단한 `pull request`를 진행하였  
고, 추후에 컨트리뷰톤이 끝나더라도 제대로된 기술적  
`pull request`를 날려볼 수 있도록 의지를 불태우고 있습  
니다! 마지막으로 어렵다고 느껴졌던 `github` 사용법을 실  
제로 프로젝트 진행하면서 도입해보니 조금 더 익숙해졌  
다는 생각이 듭니다 😊

# 오픈소스 입문의 정석, 컨트리뷰션



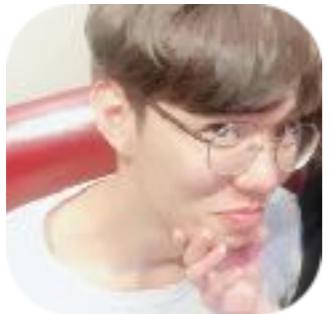
홍상원

Contributor

저에게 Git/Github는 단순히 소스코드 백업용 도구였습니다. 컨트리뷰톤에 참여하여 오픈소스에 컨트리뷰션을 하게 되면서 Git/Github는 협업을 위한 필수적인 도구임을 몸소 깨닫게 되었습니다.

uftrace에 첫 commit을 시작으로 Linux Kernel perf에 첫 patch mail을 보내기까지의 모든 과정은 컨트리뷰톤에 참여하지 못했더라면 얻을 수 없는 성과였습니다. 이번을 계기로 컨트리뷰션의 과정을 이해하게 되었고 컨트리뷰션에 대한 두려움이 사라지게 되었습니다. 컨트리뷰톤을 발판으로 삼아 앞으로 더욱 훌륭한 컨트리뷰터가 되도록 하겠습니다.

# 가장 재밌는 SNS; Github



홍한석

Contributor

버전 관리 이상의 협업 도구로써 Git/Github의 강력함을 경험했습니다! 😍

uftrace를 첫 컨트리뷰션으로 시작해, 다른 오픈소스에도 번역 작업 PR을 보냈어요. 🚢

코드 commit을 위해 공부한 DWARF 파일에 대한 내용도 도움이 많이 됐습니다. 😎

고수 개발자의 훌륭한 코드도 볼 수 있어 앞으로의 Github를 통한 오픈소스 컨트리뷰션 활동이 기대됩니다!



# Thank you!