

Linux Kernel – perf

오픈소스 개발참여방법과 개발과정의 이해

KOSSCON 2016

Taeung Song

미래부 KOSS LAB. – Software Engineer

taeung@kossilab.kr

송태웅 (Taeung Song, <https://github.com/taeung>)

- 미래창조과학부 KOSS Lab. Software Engineer
- Linux Kernel Contributor 활동 중 (perf)

강의활동


- SK C&C Git/Github 사내 교육영상 제작
- 서강대 , 아주대 , OSS 포럼 등 Git/Github 강의
- 국민대 , 이화여대 등 Linux perf, Opensource 참여 관련 시간강사 활동

Contents

1. Linux Kernel 에 **PATCH** 제출하는 방법
2. 리눅스 성능분석도구 **perf** 소개
3. 최근 작업중인 **PATCH set** 소개

Linux Kernel 공식 git repository

https://git.kernel.org



Kernel.org git repositories

Git repositories hosted at kernel.org

index

search

Name	Description	Owner	Idle	Links
bluetooth				
bluez-hcidump.git	Bluetooth packet analyzer	Marcel Holtmann	3 years	summary log tree
bluez.git	Bluetooth protocol stack for Linux	Marcel Holtmann	26 hours	summary log tree
obexd.git	OBEX Server	Marcel Holtmann	3 years	summary log tree
sbc.git	SBC library	holtmann	17 months	summary log tree
boot				
dracut/dracut.git	dracut - Initramfs generator using udev	Harald Hoyer	8 days	summary log tree
efilinux/efilinux.git	The efilinux UEFI boot loader	Matt Fleming	21 months	summary log tree
syslinux/syslinux.git	The Syslinux boot loader suite	Syslinux workgroup	18 months	summary log tree
devel				
pahole/pahole.git	Pahole and other DWARF utils	Arnaldo Carvalho de Melo	10 days	summary log tree
sparse/chrisl/sparse.git	Chris Li's sparse repository.	Christopher Li	5 weeks	summary log tree
sparse/sparse.git	C semantic parser	Christopher Li	13 months	summary log tree
docs				
kernel/kernel-docs.git	Kernel Documentation tree	Doc Group	2 years	summary log tree
kernel/ksmap.git	Kernel.org keysign map source	Kernel.org users	4 days	summary log tree
kernel/website.git	Kernel.org website source	Doc Group	13 days	summary log tree
man-pages/man-pages.git	Linux man pages Sections 2, 3, 4, 5, and 7	Michael Timothy Kerrisk	9 days	summary log tree
man-pages/website.git	Website files for /doc/man-pages	Michael Timothy Kerrisk	2 weeks	summary log tree
editors				
uemacs/uemacs.git	Micro-emacs	Linus Torvalds	16 months	summary log tree
fs				
ext2/e2fsprogs.git	Ext2/3/4 filesystem userspace utilities	Theodore T'so	3 days	summary log tree
ext2/xfstests-bld.git	Build framework and autorun scripts for xfstests	Theodore T'so	4 weeks	summary log tree
fat/fatattr/fatattr.git	FAT attribute set utility	H. Peter Anvin	7 years	summary log tree
fuse/dbfs.git	FUSE fs w/ Berkeley DB backend.	Jeff Garzik	7 years	summary log tree
fuse/fuse-ext2.git	FUSE ext2 filesystem driver.	Jeff Garzik	10 years	summary log tree
squashfs/squashfs-tools.git	squashfs tools development	Phillip Lougher	4 months	summary log tree
xfs/dmapi-dev.git	Data Management API runtime environment	Christoph Hellwig	5 years	summary log tree
xfs/xfs-documentation.git	XFS AsciiDoc Documentation tree	XFS FS Group	3 months	summary log tree

https://git.kernel.org/cgit/linux/kernel/git/tip/tip.git/

Kernel 소스 받기

git clone git://git.kernel.org/pub/scm/linux/kernel/git/tip/tip.git

또는 실습시간에 USB / Download URL 로 받기

<https://www.dropbox.com/sh/bm054rptk5q45g4/AADRPBSymi4PT4inY-9LD9Aza?dl=1&pl=1>

Git 상태 점검 및 정리

```
# cd linux-perf
```

```
# git status
```

파일 mode 변경으로 인해 변경사항이 생겼다면 ..

```
# git reset --hard master
```

vi MAINTAINERS

... 생략 ...

9425 PERFORMANCE EVENTS SUBSYSTEM

9426 M: Peter Zijlstra <peterz@infradead.org>
9427 M: Ingo Molnar <mingo@redhat.com>
9428 M: Arnaldo Carvalho de Melo <acme@kernel.org>
9429 R: Alexander Shishkin <alexander.shishkin@linux.intel.com>
9430 L: linux-kernel@vger.kernel.org
9431 T: git git://git.kernel.org/pub/scm/linux/kernel/git/tip/tip.git perf/core
9432 S: Supported
9433 F: kernel/events/*
9434 F: include/linux/perf_event.h
9435 F: include/uapi/linux/perf_event.h
9436 F: arch/*/kernel/perf_event*.c
9437 F: arch/*/kernel/*/perf_event*.c
9438 F: arch/*/kernel/*/perf_event*.c
9439 F: arch/*/include/asm/perf_event.h
9440 F: arch/*/kernel/perf_callchain.c
9441 F: arch/*/events/*
9442 F: tools/perf/

... 생략 ...

```
# cd tools/perf
```

```
# vi Documentation/tips.txt
```

임의 수정 : 자기 이름 추가 하기

git status

On branch master

Your branch is up-to-date with 'origin/master'.

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git checkout -- <file>..." to discard changes in working directory)

modified: Documentation/tips.txt

no changes added to commit (use "git add" and/or "git commit -a")

git diff

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

b/tools/perf/Documentation/tips.txt

index 8a6479c..b93fb8d 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 srclin

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

+Taeung

Commit 만들기

```
# git commit -asm "perf tools: Add a tip to tips"  
[master ee8ae28] perf tools: Add a tip to tips  
1 file changed, 1 insertion(+)
```

commit 을 PATCH 파일로 만들고 code style 검사

git format-patch -1

./scripts/checkpatch.pl --no-tree <.patch 파일명 >

Git sendmail 설정

```
# cat ~/.gitconfig
```

```
... 생략 ...
```

```
[sendemail]
```

```
smtpserver = smtp.gmail.com
```

```
smtpserverport = 587
```

```
smtpuser = 본인메일 @gmail.com
```

```
smtpencryption = tls
```

```
... 생략 ...
```

예제 PATCH mail 전송 실습

```
# sudo apt-get install git-email
```

troubleshooting) gmail 에서 보안설정 변경

```
# git send-email --to taeung@kosslab.kr *.patch
```

예제 PATCH mail 전송 실습

cat <.patch 파일명 >

내용을 이용해서 mail 클라이언트 프로그램 이용가능

실제 PATCH mail 전송 예시

```
# git send-email --confirm=never \  
--to "Arnaldo Carvalho de Melo <acme@kernel.org>" \  
--cc "linux-kernel@vger.kernel.org" \  
--cc "Jiri Olsa <jolsa@kernel.org>" \  
--cc "Namhyung Kim <namhyung@kernel.org>" \  
--cc "Ingo Molnar <mingo@kernel.org>" \  
--cc "Peter Zijlstra <peterz@infradead.org>" \  
--cc "Alexander Shishkin <alexander.shishkin@linux.intel.com>" \  
--cc "Masami Hiramatsu <mhiramat@kernel.org>" \  
--cc "Wang Nan <wangnan0@huawei.com>" *.patch
```


<https://lkml.org/lkml/2016/3/27/48>

Date Sun, 27 Mar 2016 13:16:26 +0200

From Jiri Olsa <>

Subject Re: [PATCH] perf config: Tidy up the code setting buildid dir
share 0
share 0

On Thu, Mar 24, 2016 at 04:49:33PM +0900, Taeung Song wrote:

> Add new perf_buildid_config() into perf_default_config,
> bring set_buildid_dir() next to perf_default_config,
> rename some variable name as more readable name and etc
> in order to clean up code about buildid dir.
>
> Cc: Jiri Olsa <jolsa@kernel.org>
> Cc: Namhyung Kim <namhyung@kernel.org>
> Cc: Wang Nan <wangnan0@huawei.com>
> Signed-off-by: Taeung Song <treeze.taeung@gmail.com>
> ---
> tools/perf/perf.c | 3 +--
> tools/perf/util/config.c | 57 ++++++-----
> 2 files changed, 23 insertions(+), 37 deletions(-)

though it's failry simple change we try to **separate changes**

seems like **3 independent changes**:

- perf.c hunk change
- buildid_dir_command_config/perf_buildid_config rework
- set_buildid_dir fix

thanks,
jirka

<https://lkml.org/lkml/2016/3/27/77>

Subject **Re: [PATCH] perf config: Tidy up the code setting buildid dir**

From **Taeung Song <>**

Date **Mon, 28 Mar 2016 00:41:29 +0900**

share 0

share 0

Hi, jirka

Thank you for your review :-)

On 03/27/2016 08:16 PM, Jiri Olsa wrote:

> On Thu, Mar 24, 2016 at 04:49:33PM +0900, Taeung Song wrote:

... 생략 ...

>> tools/perf/util/config.c | 57 ++++++-----

>> 2 files changed, 23 insertions(+), 37 deletions(-)

>

> though it's failry simple change we try to separate changes

>

> seems like 3 independent changes:

>

> - perf.c hunk change

> - buildid_dir_command_config/perf_buildid_config rework

> - set_buildid_dir fix

>

You mean it is needed to separate this patch as 3 part?

I got it.

I'll resend the patchset.

Thanks,
Taeung

<https://lkml.org/lkml/2016/3/28/373>

Date Mon, 28 Mar 2016 21:52:42 +0200

From Jiri Olsa <>

Subject Re: [PATCH 1/3] perf config: Remove duplicated the code calling set_buildid_dir

share 0

share 0

On Mon, Mar 28, 2016 at 02:22:18AM +0900, Taeung Song wrote:

> Cc: Jiri Olsa <jolsa@kernel.org>

> Cc: Namhyung Kim <namhyung@kernel.org>

> Signed-off-by: Taeung Song <treeze.taeung@gmail.com>

for the patchset:

Acked-by: Jiri Olsa <jolsa@kernel.org>

thanks,

jirka

> ---

> tools/perf/perf.c | 3 +--

> 1 file changed, 1 insertion(+), 2 deletions(-)

>

... 생략 ...

리눅스 성능분석도구 **perf** 란 ?

리눅스 성능분석도구 **perf** 란 ?



특정 프로그램 / 시스템 전반

리눅스 성능분석도구 **perf** 란 ?

함수단위 / 소스라인 단위



특정 프로그램 / 시스템 전반

리눅스 성능분석도구 **perf** 란 ?

함수단위 / 소스라인 단위



특정 프로그램 / 시스템 전반 with **Events Sampling**

리눅스 성능분석도구 **perf** 란 ?

함수단위 / 소스라인 단위



특정 프로그램 / 시스템 전반 with **Events** Sampling



성능 측정가능한 초점 **Focus**

리눅스 성능분석도구 **perf** 란 ?

함수단위 / 소스라인 단위

특정 프로그램 / 시스템 전반 with **Events** Sampling

성능 측정가능한 초점 **Focus** (CPU cycles,

리눅스 성능분석도구 **perf** 란 ?

함수단위 / 소스라인 단위

특정 프로그램 / 시스템 전반 with **Events** Sampling

성능 측정가능한 초점 Focus (CPU cycles, **cache-misses**)

리눅스 성능분석도구 **perf** 란 ?

함수단위 / 소스라인 단위

특정 프로그램 / 시스템 전반 with **Events** Sampling

성능 측정가능한 초점 Focus (CPU cycles, cache-misses
, **page-fault**,

리눅스 성능분석도구 **perf** 란 ?

함수단위 / 소스라인 단위

특정 프로그램 / 시스템 전반 with **Events** Sampling

성능 측정가능한 초점 Focus (CPU cycles, cache-misses
, page-fault, **system calls**, etc.)

기존에 많이 사용하는
성능 모니터링 도구들

타 성능측정도구

- top

```
top - 11:40:30 up 23:57, 5 users, load average: 1.40, 0.90, 0.63
Tasks: 382 total, 3 running, 379 sleeping, 0 stopped, 0 zombie
Cpu(s): 53.7%us, 2.7%sy, 0.0%ni, 43.4%id, 0.2%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 1024212k total, 986976k used, 37236k free, 16476k buffers
Swap: 4088500k total, 313288k used, 3775212k free, 207380k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
7087	rameshj	20	0	231m	9840	5772	R	94	1.0	1:31.81	java
2304	root	20	0	427m	89m	6548	R	9	8.9	15:27.31	X
5701	rameshj	20	0	179m	10m	6548	S	2	1.0	24:37.23	knotify4
5855	rameshj	20	0	691m	275m	20m	S	1	27.5	113:37.47	firefox-bin
5767	rameshj	9	-11	156m	3452	2936	S	1	0.3	68:22.04	pulseaudio
6089	rameshj	20	0	347m	20m	7400	S	1	2.0	81:30.14	plugin-containe
1570	rameshj	20	0	73764	10m	7168	S	0	1.1	0:08.28	konsole
1859	rameshj	20	0	579m	198m	2092	S	0	19.9	2:55.70	java
5702	rameshj	20	0	255m	13m	8024	S	0	1.3	0:26.34	plasma
5755	rameshj	9	-11	156m	3452	2936	S	0	0.3	33:29.66	pulseaudio
5887	rameshj	20	0	691m	275m	20m	S	0	27.5	8:38.24	firefox-bin
5906	rameshj	20	0	691m	275m	20m	S	0	27.5	0:54.29	firefox-bin

타 성능측정도구

- iperf

[ID]	Interval		Transfer	Bandwidth	
[4]	0.00-30.00	sec	794 MBytes	222 Mbits/sec	sender
[4]	0.00-30.00	sec	794 MBytes	222 Mbits/sec	receiver
[6]	0.00-30.00	sec	795 MBytes	222 Mbits/sec	sender
[6]	0.00-30.00	sec	795 MBytes	222 Mbits/sec	receiver
[8]	0.00-30.00	sec	786 MBytes	220 Mbits/sec	sender
[8]	0.00-30.00	sec	786 MBytes	220 Mbits/sec	receiver
[10]	0.00-30.00	sec	795 MBytes	222 Mbits/sec	sender
[10]	0.00-30.00	sec	795 MBytes	222 Mbits/sec	receiver
[12]	0.00-30.00	sec	772 MBytes	216 Mbits/sec	sender
[12]	0.00-30.00	sec	771 MBytes	216 Mbits/sec	receiver
[14]	0.00-30.00	sec	754 MBytes	211 Mbits/sec	sender
[14]	0.00-30.00	sec	754 MBytes	211 Mbits/sec	receiver
[16]	0.00-30.00	sec	756 MBytes	211 Mbits/sec	sender
[16]	0.00-30.00	sec	756 MBytes	211 Mbits/sec	receiver
[18]	0.00-30.00	sec	758 MBytes	212 Mbits/sec	sender
[18]	0.00-30.00	sec	758 MBytes	212 Mbits/sec	receiver
[20]	0.00-30.00	sec	782 MBytes	219 Mbits/sec	sender
[20]	0.00-30.00	sec	781 MBytes	219 Mbits/sec	receiver
[22]	0.00-30.00	sec	765 MBytes	214 Mbits/sec	sender
[22]	0.00-30.00	sec	765 MBytes	214 Mbits/sec	receiver
[SUM]	0.00-30.00	sec	7.57 GBytes	2.17 Gbits/sec	sender
[SUM]	0.00-30.00	sec	7.57 GBytes	2.17 Gbits/sec	receiver

iperf Done.

타 성능측정도구

- iotop

Total DISK READ: 0.00 B/s		Total DISK WRITE: 120.50 K/s				
TID	PRIO	USER	DISK READ	DISK WRITE	SWAPIN	IO>
286	be/3	root	0.00 B/s	38.87 K/s	0.00 %	0.10 %
943	be/4	root	0.00 B/s	3.89 K/s	0.00 %	0.00 %
1	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %
2	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %
1027	be/4	adamowen	0.00 B/s	0.00 B/s	0.00 %	0.00 %
4	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %
5	rt/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %
6	rt/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %
7	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %
8	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %
9	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %
10	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %
11	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %
12	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %
13	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %
14	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %
15	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %
16	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %
17	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %
18	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %
3	rt/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %
20	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %
21	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %
22	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %
23	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %
24	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %
25	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %
26	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %
27	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %
28	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %
29	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %
30	be/5	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %
31	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %
32	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %
1569	be/4	nobody	0.00 B/s	0.00 B/s	0.00 %	0.00 %
1570	be/4	nobody	0.00 B/s	0.00 B/s	0.00 %	0.00 %

개발자 입장에서 top, iperf, iotop 의 한계

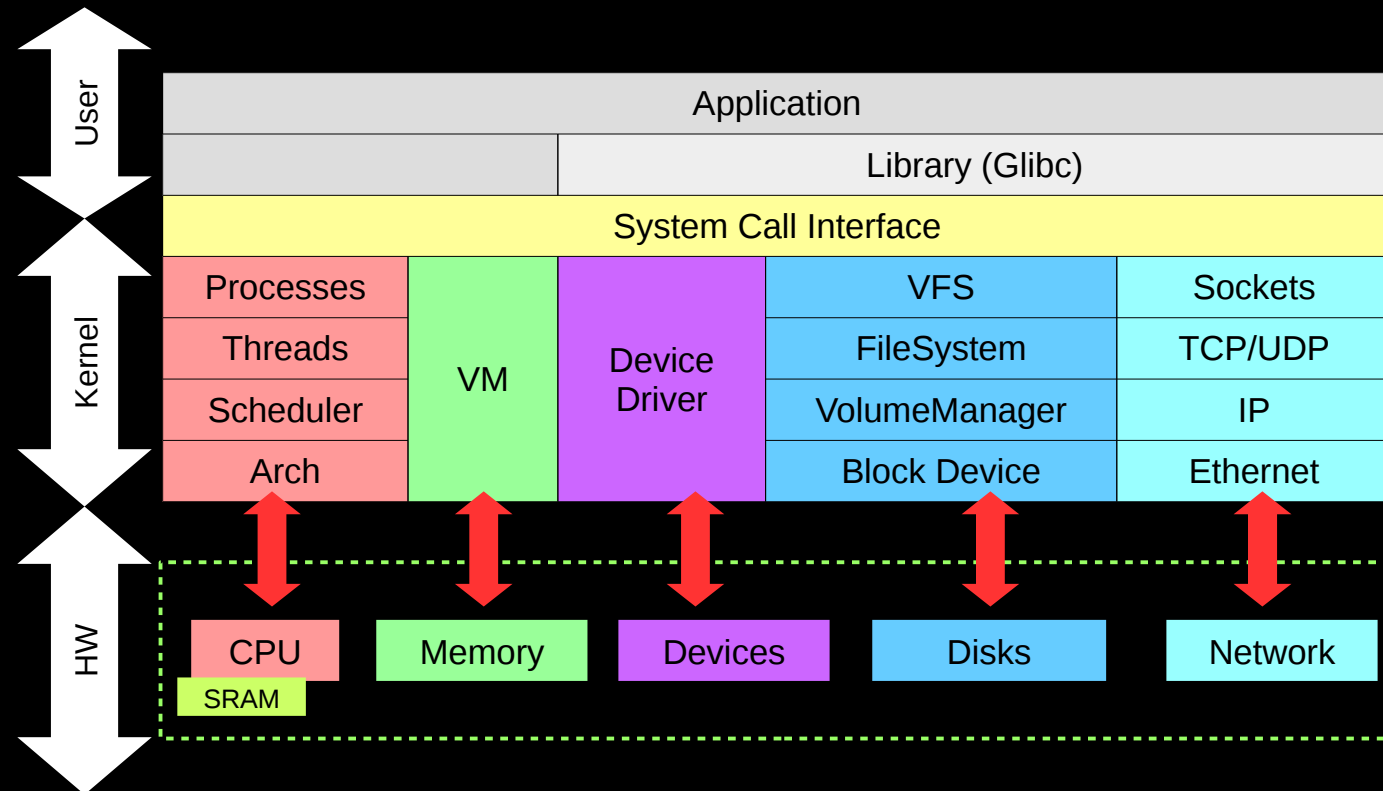
- 결과적인 CPU 점유율만 표시
- 시간당 데이터처리량만 표시
- I/O 발생 정도만 확인 가능
- 어떤 소스라인 / 함수 가 병목지점인지 알수 없음
- Disk/Network I/O 가 심하다면 상세한 진단 불가능

그런데 **perf**에서는 **각종 event** 중에서

System Layer 상에서 원하는 **성능분석 지점** (Focus) 을 **선택가능**

- **CPU cycles** 중심 성능분석
- **Block Device** 레벨 성능분석
- **File System(ext4)** 레벨 성능분석
- **Socket** 레벨 성능분석
- **Ethernet (NIC)** 레벨 성능분석
- ...

System Layer 상에서 원하는 **성능분석 지점 (Focus)** 을 **선택**가능

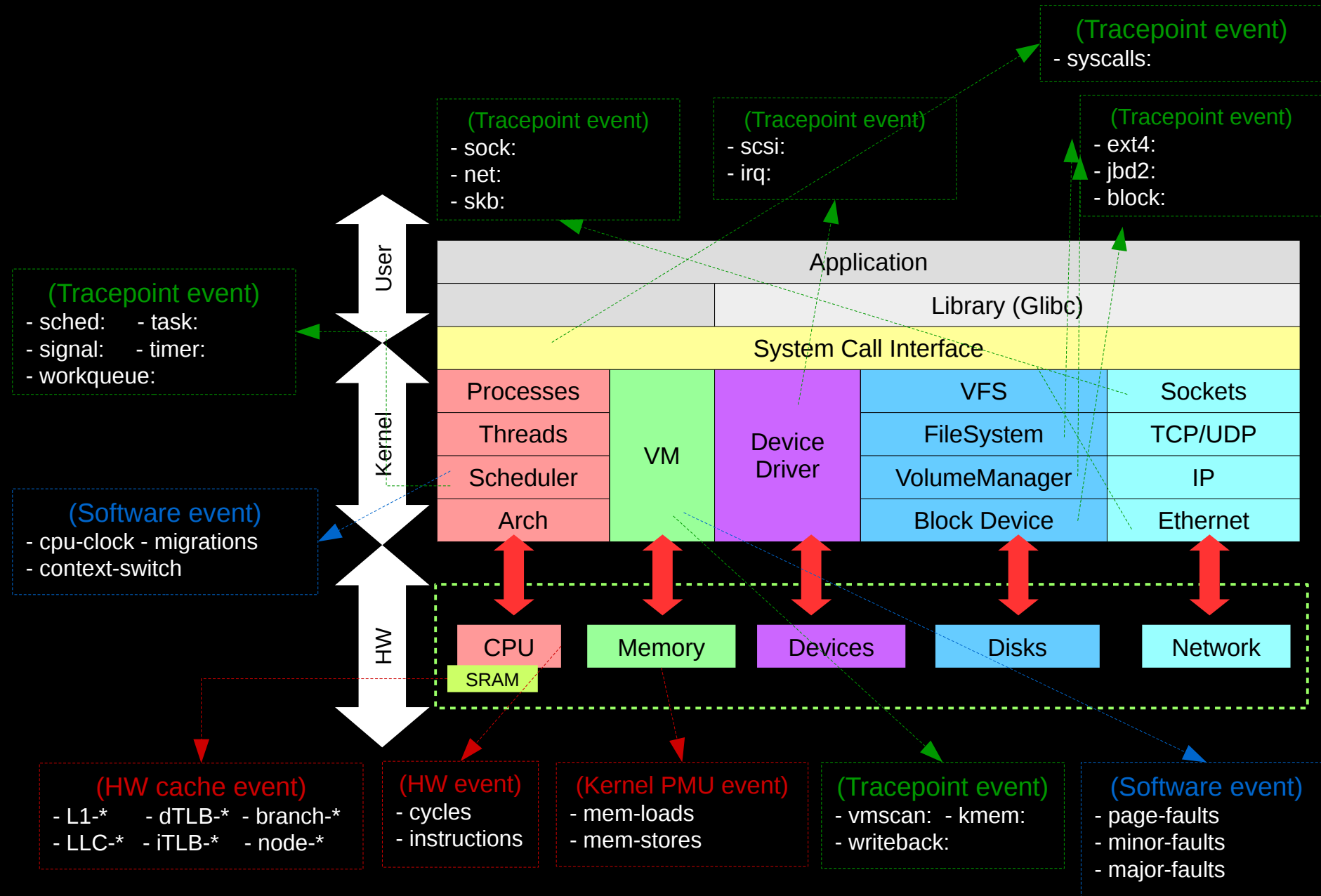


< Linux kernel 의 주요 5 가지 subsystem 기준 >

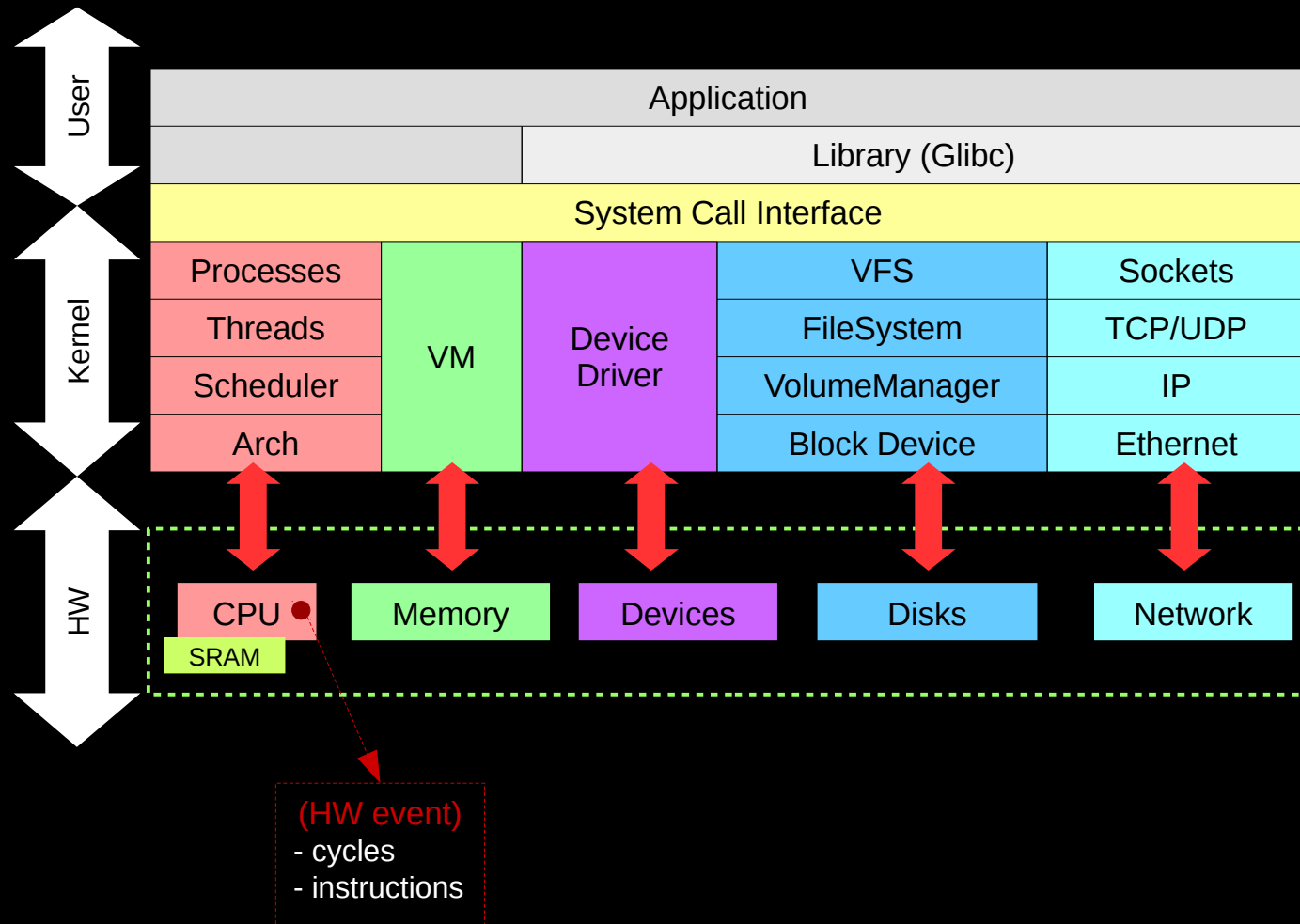
- 1) 프로세스 관리 (Process management)
- 2) 메모리 관리 (Memory management)
- 3) 디바이스 드라이버 (Device Driver)
- 4) 파일 시스템 (File System)
- 5) 네트워킹 (Networking)

http://www.makelinux.net/kernel_map/
<http://www.brendangregg.com/perf.html>

System Layer 상에서 원하는 성능분석 지점 (Focus) 을 선택가능



System Layer 상에서 원하는 **성능분석 지점 (Focus)** 을 **선택**가능



따라서 **perf** 는 단순성능측정 도구보다
한층 더 상세한 성능 분석이 가능하다

perf 의 사용목적 ?

- **Profiling**

- **Tracing**

perf 의 사용목적 ?

- **Profiling** : 병목지점 (Bottlenecks) 을 찾아내기 위해

perf 의 사용목적 ?

- **Profiling** : **병목지점** (Bottlenecks) 을 찾아내기 위해

...
21 int foo (int a, int b)
22 {
23 ...
24
25
26 }
27
28
29
30
31
32
33
...

소스코드 program.c

perf 의 사용목적 ?

- **Profiling** : **병목지점** (Bottlenecks) 을 찾아내기 위해



- 프로그램 전체에서 **어떤함수**가 **CPU** 를 많이 **사용**하는지 ?

...
21 int foo (int a, int b)
22 {
23 ...
24
25
26 }
27
28
29
30
31
32
33
...

소스코드 program.c

perf 의 사용목적 ?

- **Profiling** : **병목지점** (Bottlenecks) 을 찾아내기 위해



- 프로그램 전체에서 어떤함수가 CPU 를 많이 사용하는지 ?
- 소스코드에서 어떤 라인이 CPU 를 많이 사용하는지 ?

...
21 int foo (int a, int b)
22 {
23 ...
24
25
26 }
27
28
29
30
31
32
33
...

소스코드 program.c

perf 의 사용목적 ?

- **Profiling** : **병목지점** (Bottlenecks) 을 찾아내기 위해

↓ HW / SW Events

- 프로그램 전체에서 어떤함수가 CPU 를 많이 사용하는지 ?
- 소스코드에서 어떤 라인이 CPU 를 많이 사용하는지 ?

...
21 int foo (int a, int b)
22 {
23 ...
24
25
26 }
27
28
29
30
31
32
33
...

소스코드 program.c

perf 의 사용목적 ?

- **Profiling** : **병목지점** (Bottlenecks) 을 찾아내기 위해



HW / SW Events (CPU cycles, cache-misses ...)

- 프로그램 전체에서 어떤함수가 CPU 를 많이 사용하는지 ?
- 소스코드에서 어떤 라인이 CPU 를 많이 사용하는지 ?

...
21 int foo (int a, int b)
22 {
23 ...
24
25
26 }
27
28
29
30
31
32
33
...

소스코드 program.c

perf 의 사용목적 ?

- **Tracing** : 특정 Event 발생에 대한 경위 ,
과정 (function call graph) 을 살펴보기 위해

perf 의 사용목적 ?

- **Tracing** : 특정 Event 발생에 대한 경위 ,
과정 (function call graph) 을 **수색**하기 위해



- 특정 **커널 함수**가 왜 불러 졌을까 ?

perf 의 사용목적 ?

- **Tracing** : 특정 Event 발생에 대한 경위 ,
과정 (function call graph) 을 **수색**하기 위해



- 특정 **커널 함수**가 왜 불러 졌을까 ?
- 그 커널함수가 **호출된** (mapping 된 event 가 **발생된**) **과정** (call graph) 어땠을까 ?

perf 의 사용목적 ?

- **Tracing** : 특정 Event 발생에 대한 경위 ,
과정 (function call graph) 을 **수색**하기 위해



Tracepoint / Probe Events

- 특정 **커널 함수**가 왜 불러 졌을까 ?
- 그 커널함수가 **호출된** (mapping 된 event 가 **발생된**) **과정** (call graph) 어땠을까 ?

perf 의 사용목적 ?

- **Tracing** : 특정 Event 발생에 대한 경위 ,
과정 (function call graph) 을 **수색**하기 위해



Tracepoint / Probe Events (여러 **커널함수** , system calls, page fault ...)

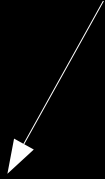
- 특정 **커널 함수**가 왜 불러 졌을까 ?
- 그 커널함수가 **호출된** (mapping 된 event 가 **발생된**) **과정** (call graph) 어땠을까 ?

Chrome 에서 파일 업로드 하는 동안

block_rq_insert 이벤트가 발생되기까지의 **과정** (call graph) 수색하면 ?

Chrome 에서 파일 업로드 하는 동안

block_rq_insert 이벤트가 발생되기까지의 **과정** (call graph) 수색하면 ?



Tracepoint Events

Chrome 에서 파일 업로드 하는 동안

block_rq_insert 이벤트가 발생되기까지의 **과정** (call graph) 수색하면 ?

```
...  
- 0.36%    0.36%  8,0 R 0 () 483479880 + 56 [chrome]
```

```
    page_fault  
    do_page_fault  
    __do_page_fault  
    handle_mm_fault  
    __do_fault  
    filemap_fault  
    __do_page_cache_readahead  
    blk_finish_plug  
    blk_flush_plug_list  
    __elv_add_request
```

```
+ 0.36%    0.36%  8,0 R 0 () 483479960 + 8 [chrome]
```

```
...
```

Chrome 에서 파일 업로드 하는 동안

block_rq_insert 이벤트가 발생되기까지의 **과정** (call graph) 수색하면 ?

```
...  
- 0.36% 0.36% 8,0 R 0 () 483479880 + 56 [chrome]
```

```
page_fault  
do_page_fault  
__do_page_fault  
handle_mm_fault  
__do_fault  
filemap_fault  
__do_page_cache_readahead  
blk_finish_plug  
blk_flush_plug_list  
__elv_add_request
```

chrome 이 **block_rq_insert** 이벤트를 발생시킴 (block I/O 요청)
(== 커널함수 **__elv_add_request** 호출함 Read 목적으로)

```
+ 0.36% 0.36% 8,0 R 0 () 483479960 + 8 [chrome]
```

```
...
```

Chrome 에서 파일 업로드 하는 동안

block_rq_insert 이벤트가 발생되기까지의 **과정** (call graph) 수색하면 ?

```
...  
- 0.36% 0.36% 8,0 R 0 () 483479880 + 56 [chrome]
```

```
page_fault  
do_page_fault  
__do_page_fault  
handle_mm_fault  
__do_fault  
filemap_fault  
__do_page_cache_readahead  
blk_finish_plug  
blk_flush_plug_list  
__elv_add_request
```

Tracepoint Events

chrome 이 **block_rq_insert** 이벤트를 발생시킴 (block I/O 요청)
(== 커널함수 **__elv_add_request** 호출함 Read 목적으로)

```
+ 0.36% 0.36% 8,0 R 0 () 483479960 + 8 [chrome]
```

```
...
```

Chrome 에서 파일 업로드 하는 동안

block_rq_insert 이벤트가 발생되기까지의 **과정** (call graph) 수색하면 ?

...

- 0.36% 0.36% 8,0 R 0 () 483479880 + 56 [chrome]

page_fault

do_page_fault

__do_page_fault

handle_mm_fault

__do_fault

filemap_fault ←

__do_page_cache_readahead

blk_finish_plug

blk_flush_plug_list

__elv_add_request

왜 __elv_add_request 가 호출이 되었나 ?

경위를 찾아 거슬러 올라가보면 ..

+ 0.36% 0.36% 8,0 R 0 () 483479960 + 8 [chrome]

...

Chrome 에서 파일 업로드 하는 동안

block_rq_insert 이벤트가 발생되기까지의 **과정** (call graph) 수색하면 ?

...
- 0.36% 0.36% 8,0 R 0 () 483479880 + 56 [chrome]

page_fault ← chrome 이 **Block I/O** 를 **요청** (block_rq_insert 이벤트 발생시킨) 한 **이유**
: **page fault** 가 발생 했기 때문에 실제 **Read** 를 요청 했다 .

do_page_fault
__do_page_fault
handle_mm_fault
__do_fault
filemap_fault
__do_page_cache_readahead
blk_finish_plug
blk_flush_plug_list
__elv_add_request

+ 0.36% 0.36% 8,0 R 0 () 483479960 + 8 [chrome]

...

최근 작업중인 PATCH set 소개

<https://lkml.org/lkml/2016/11/4/46>

cat 0000-cover-letter.patch

From 00298b6fdb0ac6e05a63ed9d108cda5ebd6f775c Mon Sep 17 00:00:00 2001

From: Taeung Song <treeze.taeung@gmail.com>

Date: Fri, 4 Nov 2016 15:42:08 +0900

Subject: [PATCH 0/6] perf config: Add support for setting and getting functionalities

Hello, :)

Add setting and getting features to perf-config.

I had worked at the related patchset <https://lkml.org/lkml/2016/2/22/38>

But I remake new this patchset for only support for read/write config file.

And There're Namhyung's requests <https://lkml.org/lkml/2016/10/24/572>

In particular, I agonized implement way for setting functionality.

I especially wonder other opinions of new perf_config_set__collect()

and bool from_system_config variable.

If someone review this patchset and give me some feedback,

I'd appreciated it. :)

Thanks,

Taeung

... 생략 ...

Cover Letter 란 ?

```
# git format-patch -7 --cover-letter --subject-prefix="PATCH v7"
```

cat 0000-cover-letter.patch

FFrom c42eb78703b28c8bcb0c515abeff683d23e9ff87 Mon Sep 17 00:00:00 2001

From: Taeung Song <treeze.taeung@gmail.com>

Date: Mon, 5 Sep 2016 14:13:03 +0900

Subject: [PATCH v7 0/7] perf config: Introduce default config key-value pairs arrays

Hello, :)

When initializing default perf config values,
we currently use values of actual type(int, bool, char *, etc.).
But I suggest using default config key-value pairs arrays.

... 생략 ...

v7:

- fix wrong handling a exception from strdup (Arnaldo)
- rebased on current acme/perf/core

v6:

- rename 'fore_back_colors' to simple 'colors' of ui_browser_colorset (Namhyung)
- remove unnecssary whitespace changes of PATCH 4/7, 5/7 (Namhyung)
- make more general macro instead of making accessor macro for each config section (Namhyung)
- rebased on current acme/perf/core

... 생략 ...

```
# git shortlog -sn --no-merges | nl | head -100
```

1	5919	H Hartley Sweeten
2	5572	Al Viro
3	4733	Takashi Iwai
4	4492	Mauro Carvalho Chehab
5	3967	Mark Brown
6	3875	Tejun Heo
7	3781	David S. Miller
8	3766	Johannes Berg
9	3481	Greg Kroah-Hartman
10	3433	Russell King

... 생략 ...

Thank you

KOSSCON 2016

Taeung Song

미래부 KOSS LAB. – Software Engineer

taeung@kosslab.kr