# **성능 측정 도구**

## **스케쥴링 분석 도구**

전화 응답 시 응답 시간 지연이 시스템 내 프로세스들간의 잦은 문맥 교환 때문으로 판단되어 전화 응답 간에 발생하는 문맥 교환 상태에 대하여 모니터링 하였다. 이때 사용한 기술은 커널의 성능 추적 메커니즘인 Ftrace 이다.

### **Ftrace**

Ftrace linux kernel 의 2.6.27 버전부터 커널 내부에 추가된 커널의 성능 측정, 디버깅을 위한 도구이다 이 도구를 이용하여 커널 함수의 성능 측정, 커널 함수의 Call-graph 확인, 문맥 교환 상태 확인 등을 확인할 수 있다. 이 중 문맥 교환 상태 확인 기능을 이용하여 터치 이벤트 발생 시의 문맥 교환 상태를 확인하였다.

#### **Ftrace 설정**

Ftrace 를 사용하기 위해서 다음과 같이 커널을 설정해줘야 한다.

1. menuconfig 에서 Kernel hacking 으로 진입한다.



1. Tracers 로 진입한다.



1. Kernel function tracer 를 enable 시킨다.



1. 커널을 새로 빌드한다.

#### **Ftrace 사용 방법**

1. Ftrace 는 Ftrace 제어 파일들을 debugfs 에 담고 있다. debugfs 는 보통 /sys/kernel/debug 에 마운트 되는데 시스템 부팅 후 만약 debugfs 가 마운트 되어 있지 않다면 다음과 같이 직접 마운트를 해야 한다.

|  |
| --- |
| # mount –t debugfs nodev /sys/kernel/debug |

1. 마운트가 완료 되면 /sys/kernel/debug/ 에 tracing 이라는 디렉토리가 새로 생성 되었을 것이다.이 디텍토리에 Ftrace 를 제어할 수 있는 파일들이 존재한다. tracing 디렉토리의 내용은 다음과 같다.

이중 우리가 사용하는 파일들은 위 그림의 점선 박스이다.

available\_tracers 파일은 사용 가능한 ftrace tracer 목록을 보여주는 파일이다. 간단하게 cat 명령어를 이용해서 현재 사용 가능한 tracer 의 목록을 확인할 수 있다. 문맥 교환 상태 확인을 위해서는 이 중 sched\_switch 를 사용하면 된다.

current\_tracer 는 tracer 를 선택하는 파일이다. tracer 를 선택하는 방법은 간단하게 위의 available\_tracers 파일에서 확인한 tracer 중 하나를 선택하여 아래의 그림처럼 echo 명령어를 이용하여 current\_tracer 파일에 써주면 된다.



그리고 현재 선택된 tracer 는 cat 을 통해서 확인할 수 있다.

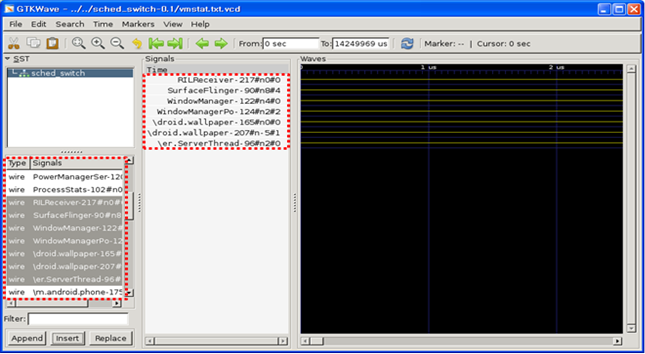
성능 측정 enable 과 disable 은 tracing\_enabled 와 tracing\_on 파일을 이용하면 된다. 각각의 파일에 숫자 0과 1 을 echo 명령어를 이용하여 써주는 방식으로 측정 시작과 중지 명령을 내릴 수 있다. 1은 측정시작을 의미하고 0은 중지를 의미한다. 일단echo 명령어를 이용하여 tracing\_enabled 파일에 한번 1 을 써주고 그 후에 tracing\_on 파일을 이용하여 시작과 중지를 반복하면 된다.



측정이 된 결과는 trace 파일에서 확인할 수 있다.

|  |  |
| --- | --- |
| **구분** | **내용** |
| 1 | Task name |
| 2 | Process Name |
| 3 | Timestamp(us) |
| 4 | <PID>:<PRIO>:<STATE>  <PRIO> : 0(high) to 98(low) , 100(high) to 139(low)  <STATE> :  R - running : wants to run, may not actually be running  S - sleep : process is waiting to be woken up (handles signals)  D - disk sleep (uninterruptible sleep) : process must be woken up (ignores signals)  T - stopped : process suspended  t - traced : process is being traced (with something like gdb)  Z - zombie : process waiting to be cleaned up  X – unknown |
| 5 | + : wake up, ==> context switch |
| 6 | <PID>:<PRIO>:<STATE> |
| 7 | Task name |

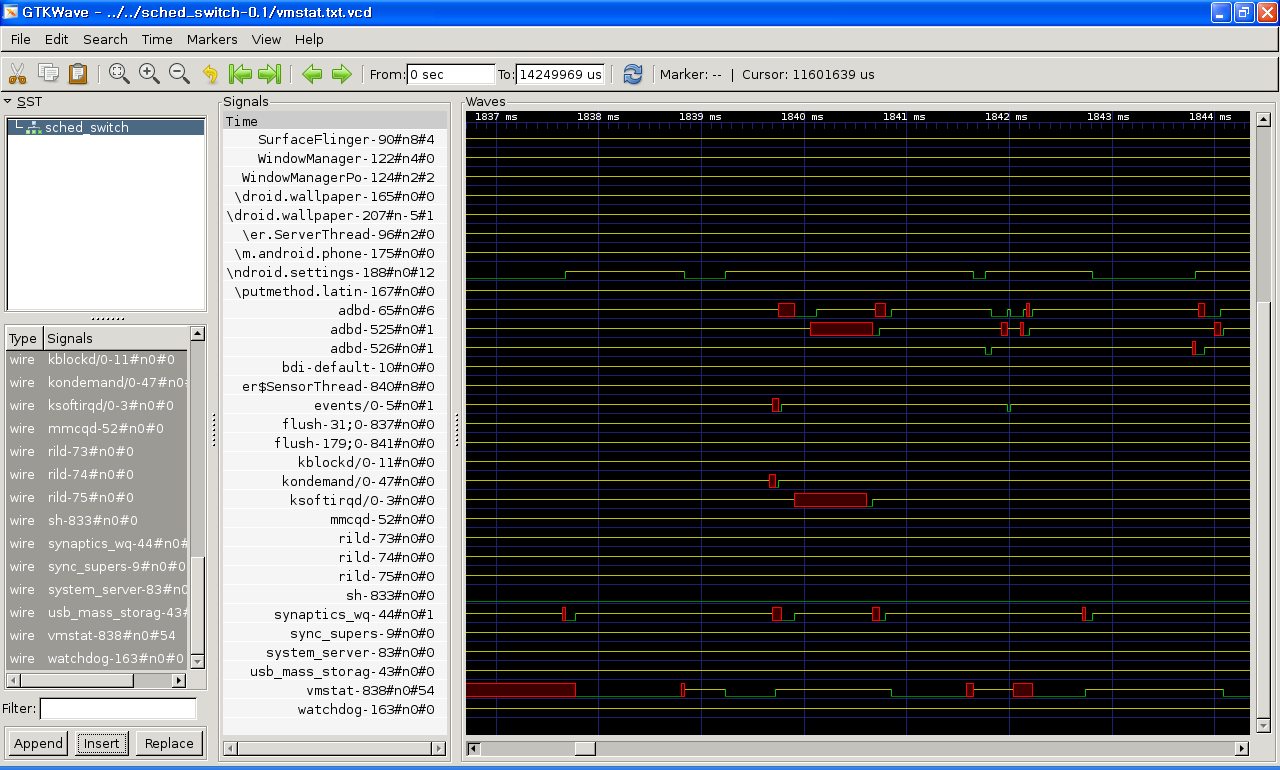
1. Visualizer 를 통한 결과 확인
   1. Visualizer 설치
      1. [http://www.om어플리케이션edia.org/images/6/6d/Sched\_switch-0.1.1.tar.gz](http://www.omappedia.org/images/6/6d/Sched_switch-0.1.1.tar.gz) 에서 해당 압축 파일을 받는다.
      2. 적당한 위치에 압축을 풀고 make 명령어로 해당 파일을 컴파일 한다. 컴파일이 완료되면 sched\_switch 라는 바이너리가 생성된다. 이것은 ftrace 를 이용하여 trace 한 문맥교환 상태 log 를 visualize 된 환경에서 볼 수 있도록 포멧을 vcd 파일로 변환 시켜주는 도구이다.
      3. GTKwave 를 설치한다. GTKwave 는 Unix 와 Window 환경에서의 wave viewer 이다.
         1. <http://gtkwave.sourceforge.net/gtkwave-3.3.20.tar.gz> 에서 해당 파일을 다운로드 한다.
         2. 적당한 위치에 압축을 해제 한다.
         3. ./configure
            1. xz 이 없다는 에러 메시지가 나올 경우 ./configure –disable-xz 로 다시 한다.
            2. make
            3. make install
2. 결과 확인
   1. sched\_switch 바이너리를 이용하여 trace 파일을 vcd 포멧으로 변환한다.
      1. ./sched\_switch <target file> <newfile.vcd> (ex. ./sched\_switch trace.txt trace.vcd)
   2. GTKwave 를 이용하여 변환된 vcd 포멧의 trace결과 데이터를 읽어서 그래프로 확인한다.
      1. GTKwave 실행
         1. gtkwave <trace.vcd>
      2. 결과 확인
         1. SST 에서 sched\_switch 를 선택 하게 되면 아래 그림의 왼쪽 빨간 박스 부분에 TASK 들의 목록이 나타난다. 그 중 확인하고 싶은 TASK 들을 선택하여 아래의 Insert 버튼을 누르게 되면 오른쪽 빨간 박스 부분에 선택한 TASK 목록들이 들어가게 된다.



* + - 1. 툴 박스의 빨간 동그라미 부분을 클릭 하면 선택 된 TASK 들에 대한 문맥 교환 상태를 한눈에 확인 할 수 있다.



자세히 확인하고 싶은 구간이 있는 경우 마우스 왼쪽 키로 특정 구간을 드래깅 하면 그 구간에 대해 좀 더 자세히 파악할 수 있다.



GTKwave 에 대한 자세한 내용은 아래의 주소에서 확인할 수 있다.

<http://sourceforge.net/projects/gtkwave/>