

Battery Historian 실습

2017. 05. 31

2017. 06. 01

2017-1 모바일 시스템 프로그래밍 01분반

한국기술교육대학교

Battery Historian

- 안드로이드 배터리 관련 정보 및 이벤트 정보를 확인할 수 있음
- 안드로이드 5.0 (Lollipop, API21) 이상의 버전 필요
- [참고] <https://github.com/google/battery-historian>



Battery Historian 설치 요약

- 본 문서에서는 Windows에서 소스코드를 이용한 설치를 진행

이미 설치된 경우 생략 가능

1. Go 언어 설치 (Golang 1.8.1 이상) <https://golang.org/doc/install>
2. Git 설치 <https://git-scm.com/downloads>
3. Python 2.7 버전 설치 (Python 3이 아님) <https://www.python.org/downloads/>
4. Java 설치 <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

5. Battery Historian 설치

1. Golang 설치

- Windows 전용 설치 파일 다운로드 및 설치

[https://storage.googleapis.com/golang/go1.8.1.windows-amd64.msi\](https://storage.googleapis.com/golang/go1.8.1.windows-amd64.msi)

2. Git 설치

- Windows 전용 설치 파일 다운로드 및 설치

<https://git-scm.com/download/win>

3. Python 2.7 설치

- Windows 전용 설치 파일 다운로드 및 설치

<https://www.python.org/ftp/python/2.7.13/python-2.7.13.amd64.msi>

- (참고) Python 3.X 버전과 Python 2.X 버전을 Windows에서 같이 사용하고 싶을 경우?
→ Anaconda Environment 사용 추천

<https://www.continuum.io/downloads>

4. 자바 런타임 환경(JRE) 설치

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

- 예) jre-8u131-windows-x64.exe

5. Battery Historian 설치

- 환경변수(시스템 변수) 등록

변수 이름	변수 값	예시
GOPATH	Go 작업 경로	C:\Users\JongSeop\go
GOBIN	Go 실행 파일 경로	%GOPATH%\bin
GOROOT	Go 설치 경로	C:\Go\
Path	(추가) Go 실행 파일 경로	%GOBIN%

5. Battery Historian 설치

- Battery Historian 소스 다운로드

- 1) 콘솔(CMD) 창을 열고
- 2) cd 명령어를 이용해서 Go 작업 경로 (GOPATH)로 이동 (ex: cd %GOPATH)
- 3) 다음 명령어를 입력하여 Battery Historian 소스 코드를 다운로드

```
go get -d -u github.com/google/battery-historian/...
```

- 4) cd 명령어를 이용하여 다운로드 된 경로로 이동

```
cd %GOPATH%/src/github.com/google/battery-historian
```

- 5) 설치 파일 실행

```
go run setup.go
```

- 6) Battery Historian 실행 (기본 포트=9999)

```
go run cmd/battery-historian/battery-historian.go
```

- 7) 인터넷 브라우저를 이용해서 Battery Historian 서버에 접속
<http://localhost:9999>

Adb (Android Debug Bridge)를 활용하여 기록 추출

- CMD 창에서 adb.exe 를 활용
 - 위치: (안드로이드SDK 설치 경로)/sdk/platform-tools/

- 안드로이드 기기가 연결된 상태에서
 - 콘솔에 다음 명령어를 입력 → 상태 기록 초기화

```
adb shell dumpsys batterystats --reset
```

- 콘솔에 다음 명령어를 입력 → Wakelock 활성화

```
adb shell dumpsys batterystats --enable full-wake-history
```

- 콘솔에 다음 명령어를 입력 → bugreport.zip 이라는 이름으로 현재 경로에 기록 추출

```
adb bugreport bugreport.zip
```


추출된 기록을 Battery Historian을 이용하여 분석

- 서버에 Bugreport.zip 파일을 업로드


Battery Historian


Upload Bugreport


Both .txt and .zip bug reports are accepted.

 Browse

Choose a Bugreport File

 Kernel Wakesource Trace

 Power Monitor File

 Switch to Bugreport Comparison

실습 과제 (온라인교육 사이트 과제란 참고)

06/01 23:59 까지

문의: 조교 박종섭 (pjs0329s@koreatech.ac.kr)

* Battery Historian 실습 과제 (개인별 과제)

<https://github.com/google/battery-historian>

아래 3가지 항목에 대해서 Battery Historian을 사용하여 리소스 사용 로그 파일을 추출하고, 관련 정보를 표시하는 timeline 차트 및 자원 사용 통계 정보를 확인해본다.

- 애플리케이션을 10분 정도 혹은 그 이상 실행한 후 로그 파일을 추출하세요.

1. Wifi scan, wakelock 사용 분석 비교

- MSP10WifiScanTimer wakelock 사용한 것

- MSP10WifiScanTimer wakelock 사용하지 않은 것

2. wakelock, sensor 사용 분석 비교 (duty cycling, adaptive duty cycling 차이 비교)

- MSP13StepMonitor2

- MSP15DCStepMonitor

- MSP16ADCStepMonitor

3. GPS, wakelock 사용 분석

- MSP17HSLocationTracking

* 05/31, 06/01 실습 시간에 위 항목에 대해서 박종섭 조교에게 확인받으세요.

(확인을 받지 않으면 점수 없음. 수업 시간에 완료하지 못 한 경우 나중에라도 따로 확인받을 것.)

* 결과물 제출

로그 파일 (bugreport.txt) - 총 6개 (각 항목별로 구분이 쉽게 되도록 파일 이름을 지정할 것)

timeline 그래프 캡처 이미지 파일 - 총 6개 (각 항목별로 구분이 쉽게 되도록 파일 이름을 지정할 것)

자원 사용 분석 비교 내용에 대해 간략히 기술한 보고서 (1, 2, 3각 항목 별로 자원 사용에 어떤 차이가 있는지 확인한 내용을 간략히 쓸 것)

* 주의사항

2, 3번의 경우는 행동(움직임이 있는지 없는지)에 따라 자원 사용 패턴이 달라지므로 10분간 애플리케이션을 실행하면서 자신이 어떻게 행동했는지도 같이 고려해야 하고, 이것도 보고서에서 언급해야 함.

예를 들어, 3번의 경우 10분 중에 앞에 3분은 움직이고, 중간 5분은 안 움직이고, 나머지 2분은 움직이는 상황이었기 때문에, wakeloc k, GPS 사용이 이렇게 이렇게 나오더라.