

12장 안드로이드 개발 지원 환경 및 분석





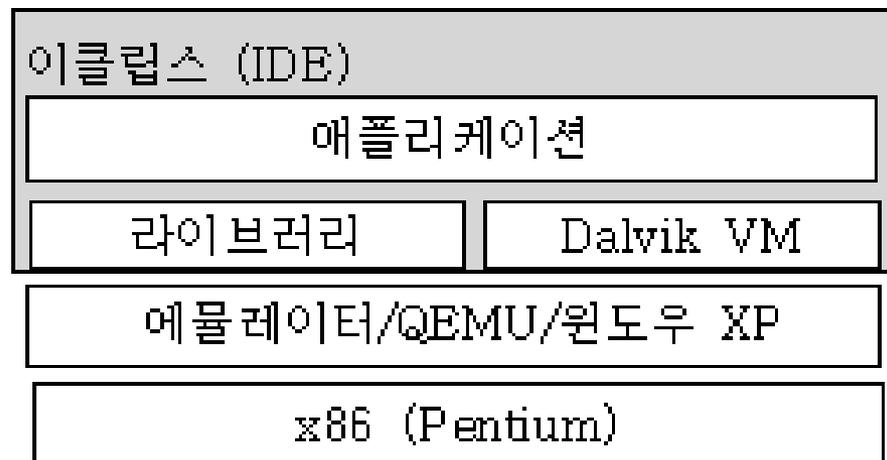
◎ 목차

- 안드로이드 개발 환경
- 안드로이드 개발 지원 도구
- 에뮬레이터에 안드로이드용 busybox 설치
- 파일 추출 실습
- 안드로이드에 SD 카드를 가상장치로 연결하기

12.1 안드로이드 개발 환경



- ⊙ 계층 구조의 안드로이드 플랫폼을 어떻게 빌드 할 것인가?
- ⊙ X86, 윈도우 기반 애플리케이션 -> ARM, 리눅스 기반의 타킷에 적재 -> 마켓에 배포
- ⊙ 윈도우 기반 개발 환경의 구성 및 동작 과정
 - 이클립스 (IDE) 지원
 - QEMU 가상머신, ARM과 리눅스 지원 에뮬레이터

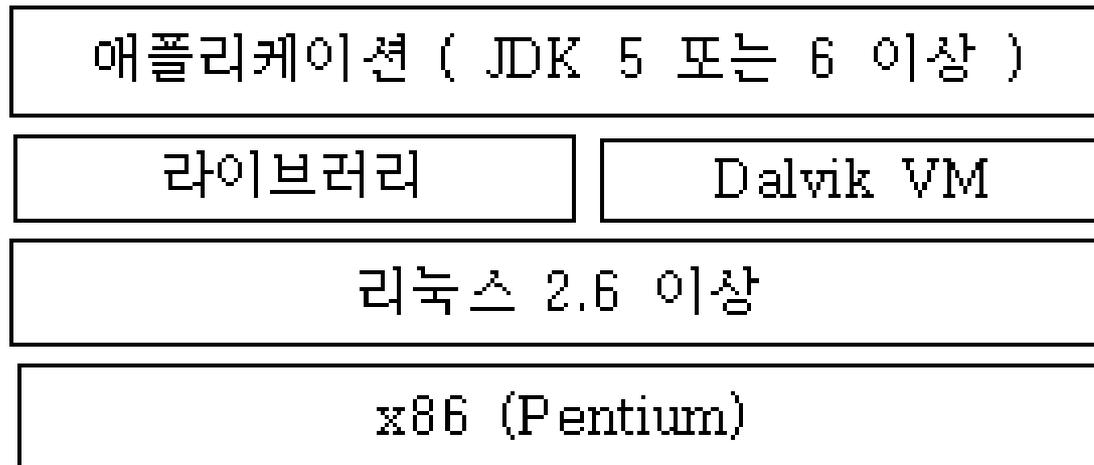


[그림 12-1] 윈도우 기반 개발환경

12.1 안드로이드 개발 환경



- ◎ X86, 리눅스 기반의 안드로이드 개발 환경
 - 안드로이드 커널 빌드에 활용
 - 하드웨어 드라이버 개발에 활용

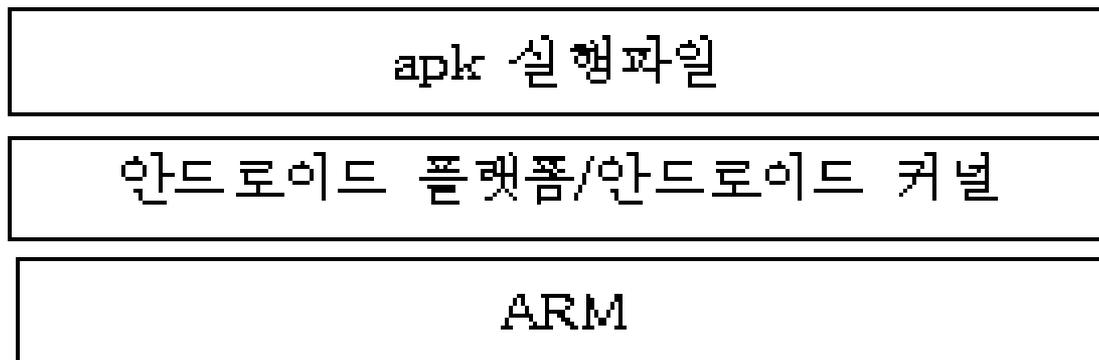


[그림 12-2] 리눅스 기반의 안드로이드 개발 환경

12.1 안드로이드 개발 환경



- ◎ ARM 기반 실행 환경
 - 실제 모바일 장치
 - 실제 타겟에 커널, 플랫폼, 파일시스템 적재 방법
 - apk 파일만 실행할 수 있는 환경 적재
 - 자바, 리눅스 빌드 도구 필요 없음



[그림 12-3] ARM 기반 실행환경

12.2 안드로이드 개발 지원 도구



- ◎ 주 개발도구
 - 자바 코드 에디터, 컴파일러, 빌더
- ◎ 개발 지원도구
 - 디버거, 파일관리, 인증, 배포 도구
 - aapt
 - adb
 - mksdcard
 - ddms
 - sqlite



[그림 12-4] 개발지원도구

12.2 안드로이드 개발 지원 도구



◎ adb

- 에뮬레이터 상태 조회 및 관리
- 리눅스 셸 명령어 구동
- 파일 복사 및 이동
- Port forward
- 클라이언트/서버(adbd)로 동작, TCP/IP 통신
- 서버는 에뮬레이터에서 백그라운드 프로세스로 동작
- Adb 클라이언트로 에뮬레이터와 연결

```
adb [-d|-e|-s <serialNumber>] <command>
```

12.2 안드로이드 개발 지원 도구



◎ adb 실습 [실습 12-1]

- (1) 윈도우의 명령창 command shell을 띄운다.
- (2) adb 프로그램이 있는 디렉토리(D:\android\android-sdk\tools)로 이동한다.
- (3) adb를 실행한다.

A screenshot of a Windows Command Shell window. The title bar reads "C:\ Command Shell". The window content shows the following text:

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Program Files\Windows Resource Kits\Tools>cd D:\android\android-sdk\tools

C:\Program Files\Windows Resource Kits\Tools>adb shell
* daemon not running. starting it now *
* daemon started successfully *
error: device not found

C:\Program Files\Windows Resource Kits\Tools>
```

[그림 12-5] 에뮬레이터가 동작하지 않고 있는 경우 오류 메시지

12.2 안드로이드 개발 지원 도구



◎ adb 실습 [실습 12-1]

(4) 이클립스에서 com.corea.Android 패키지를 실행시켜 에뮬레이터를 구동시킨다. 실행 과정을 [그림 12-6]처럼 이클립스 콘솔화면에서 관찰할 수 있다. 안드로이드가 런치(Launch) 하면서, adb 데몬, Android 애플리케이션도 런치되고, myAVD라는 가상장치를 기반으로 에뮬레이터도 동작한다. 에뮬레이터에 com.corea.Android 애플리케이션이 업로드되어 동작한다.

The screenshot shows the Eclipse IDE's console window. The title bar includes 'Problems', '@ Javadoc', 'Declaration', and 'Console'. The console output shows the following sequence of events:

```
android] -----
android] Android Launch!
android] adb is running normally.
android] Performing com.corea.Android.Android activity launch
android] Automatic Target Mode: launching new emulator with compatible AVD 'myAVD'
android] Launching a new emulator with Virtual Device 'myAVD'
android] New emulator found: emulator-5554
android] Waiting for HOME ('android.process.acore') to be launched...
android] HOME is up on device 'emulator-5554'
android] Uploading android.apk onto device 'emulator-5554'
android] Installing android.apk...
android] Success!
android] Starting activity com.corea.Android.Android on device
android] ActivityManager: Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER]
```

[그림 12-6] myAVD 에뮬레이터에 Android 패키지 동작 과정

12.2 안드로이드 개발 지원 도구



◎ adb 실습 [실습 12-1]

(5) 다시 adb 프로그램을 실행시킨다. 에뮬레이터의 adbd 서버가 동작됨으로 adb로 연결 되어 리눅스 셸 명령어를 사용할 수 있다. 'adb shell'를 입력하면 [그림 12-7]처럼 셸 프롬프트shell prompt가 >에서 #로 변경된다. DOS 모드에서 ubuntu 리눅스의 셸 모드에 들어간 것이다. ls 같은 리눅스 셸 명령어를 사용할 수 있다. 안드로이드는 일반 리눅스 셸 명령어를 최적화한 극히 일부 명령어만 지원한다.

A screenshot of a Windows Command Shell window. The title bar reads "C:\ Command Shell - adb shell". The command prompt shows the path "C:\Program Files\Windows Resource Kits\Tools>" followed by the command "adb shell". The prompt has changed from ">" to "#", indicating a successful connection to a Linux shell. There is a small cursor after the "#".

```
C:\Program Files\Windows Resource Kits\Tools>adb shell
#
```

[그림 12-7] 원격 셸 접속

12.2 안드로이드 개발 지원 도구



- [실습 12-2] 호스트 컴퓨터에 현재 연결되어 동작중인 에뮬레이터 및 단말기 같은 장치목록을 검색하는 실습
 - (1) 이클립스 'Windows' 메뉴의 'Android SDK and AVD Manager'를 선택한다. 그리고 Virtual Devices를 선택하여 새로운 가상장치를 만들기 위해 'New'를 선택한다. [그림 12-8] 처럼 API Level 3의 'test'라는 가상장치를 등록한다.



[그림 12-8] test라는 가상장치를 등록

12.2 안드로이드 개발 지원 도구



- [실습 12-2] 호스트 컴퓨터에 현재 연결되어 동작중인 에뮬레이터 및 단말기 같은 장치목록을 검색하는 실습
 - (2) 생성한 'test'라는 가상장치 에뮬레이터를 [그림 12-9]처럼 'emulator -avd test' 명령으로 1개 더 시동시킨다.

A screenshot of a Windows Command Shell window. The title bar reads "C:\ Command Shell - emulator -avd test". The command prompt shows the path "C:\Program Files\Windows Resource Kits\Tools>" followed by the command "emulator -avd test". The output is "emulator: emulator window was out of view and was recentred".

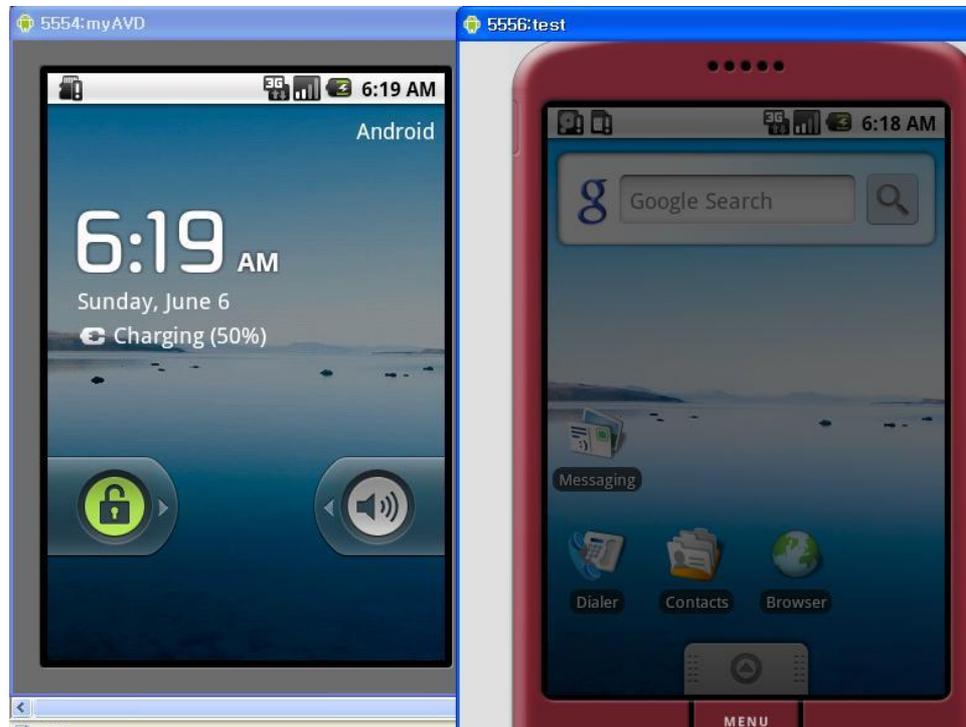
```
C:\Program Files\Windows Resource Kits\Tools>emulator -avd test
emulator: emulator window was out of view and was recentred
```

[그림 12-9] test라는 이름의 가상장치를 시동

12.2 안드로이드 개발 지원 도구



- [실습 12-2] 호스트 컴퓨터에 현재 연결되어 동작중인 에뮬레이터 및 단말기 같은 장치목록을 검색하는 실습
 - (3) 두 개의 에뮬레이터가 [그림 12-10]처럼 호스트에서 동작중인 것을 확인한다.



[그림 12-10] 두 개의 에뮬레이터가 동작

12.2 안드로이드 개발 지원 도구



- [실습 12-2] 호스트 컴퓨터에 현재 연결되어 동작중인 에뮬레이터 및 단말기 같은 장치목록을 검색하는 실습
 - (4) 또 다른 명령 창을 열고, adb devices 을 입력한다. [그림 12-11]처럼 두 개의 에뮬레이터가 동작중인 것을 확인한다.

A screenshot of a Windows Command Shell window. The title bar reads 'C:\ Command Shell'. The window content shows the following text:

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Program Files\Windows Resource Kits\Tools>adb devices
List of devices attached
emulator-5554    device
emulator-5556    device

C:\Program Files\Windows Resource Kits\Tools>
```

[그림 12-11] adb devices 명령 사용

12.2 안드로이드 개발 지원 도구



○ [실습 12-3] 복수 에뮬레이터 고급 활용하기

(1) `adb -s emulator-5556 shell` 을 입력한다. [그림 12-12]는 5556번 test 에뮬레이터 가상장치에 셸 모드로 연결한 상태를 보여준다.

A screenshot of a Windows Command Shell window. The title bar reads "C:\> Command Shell - adb -s emulator-5556 shell". The command prompt shows the path "C:\Program Files\Windows Resource Kits\Tools>" followed by the command "adb -s emulator-5556 shell". The output is a shell prompt "# _".

```
C:\> Command Shell - adb -s emulator-5556 shell
C:\Program Files\Windows Resource Kits\Tools>adb -s emulator-5556 shell
# _
```

[그림 12-12] 셸 모드 연결하기

12.2 안드로이드 개발 지원 도구



○ [실습 12-3] 복수 에뮬레이터 고급 활용하기

(2) 셸 모드에서 'ls -l' 명령어를 입력한다. adb를 통하여 셸 명령어를 사용해보면 안드로이드가 리눅스 기반이란 것을 이해할 수 있다. 셸 명령어 "ls -l"을 사용하여 루트root 디렉토리 구조를 살펴보고 [그림 12-13]처럼 리눅스 파일시스템을 관찰을 실습한다.

```

C:\Program Files\Windows Resource Kits\Tools>adb -s emulator-5556 shell
# ifconfig eth0
ifconfig eth0
eth0: ip 10.0.2.15 mask 255.255.255.0 flags [up broadcast running multicast]
# ls -l
ls -l
drwxrwxrwt root      root      2010-06-06 06:13 sqlite_stmt_journals
drwxrwx--- system  cache    2010-03-27 08:24 cache
d----- system  system   2010-06-06 06:13 sdcard
lrwxrwxrwx root      root      2010-06-06 06:13 etc -> /system/etc
drwxr-xr-x root      root      2009-07-01 00:24 system
drwxr-xr-x root      root      1970-01-01 00:00 sys
drwxr-x--- root      root      1970-01-01 00:00/sbin
dr-xr-xr-x root      root      1970-01-01 00:00/proc
-rwxr-x--- root      root      9075 1970-01-01 00:00/init.rc
-rwxr-x--- root      root      1677 1970-01-01 00:00/init.goldfish.rc
-rwxr-x--- root      root      106568 1970-01-01 00:00/init
-rw-r--r-- root      root      118 1970-01-01 00:00/default.prop
drwxrwx--x system  system   2010-03-27 08:24 data
drwx----- root      root      1970-01-01 00:00 root
drwxr-xr-x root      root      2010-06-06 06:14 dev
#
  
```

[그림 12-13] 리눅스 파일시스템 관찰

12.2 안드로이드 개발 지원 도구



○ [실습 12-3] 복수 에뮬레이터 고급 활용하기

(3) 5554 번 에뮬레이터 명령창을 띄운다.

(4) 셸 모드에서 `ifconfig eth0` 를 입력한다. 셸 모드에서 5554 번 에뮬레이터의 IP 번호 설정 상태를 확인하기 위하여 명령창에서 네트워크 해석 도구인 `netstat -anob` 입력한다. IP 주소가 10.0.2.15 로 [그림 12-14]처럼 확인된다.

A screenshot of a Windows Command Shell window titled "Command Shell - adb -s emulator-5554 shell". The window shows the following text:

```
# ifconfig eth0
ifconfig eth0
eth0: ip 10.0.2.15 mask 255.255.255.0 flags [up broadcast running multicast]
#
```

[그림 12-14] 에뮬레이터 IP 주소 확인

12.2 안드로이드 개발 지원 도구



○ [실습 12-3] 복수 에뮬레이터 고급 활용하기

(5) 셸 모드에서 `netstat -anob` 를 입력한다. 5554 번 에뮬레이터에서 동작 중인 TCP 통신 프로토콜 연결 상태를 확인하는 명령어이다. 모든 adb 클라이언트는 서버와의 통신에 TCP 5037 포트를 바인딩하여 사용하며, [그림 12-15] 처럼 에뮬레이터 서버 5555번 포트로 adb와 TCP 연결이 설정이 이루어진 것을 확인할 수 있다.

```

C:\Program Files\Windows Resource Kits\Tools>adb -s emulator-5554 shell
# netstat -anob
netstat -anob
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.1:5037          0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:5555           0.0.0.0:*               LISTEN
tcp        0      0 10.0.2.15:5555         10.0.2.2:2616          ESTABLISHED
# _
  
```

[그림 12-15] TCP 연결 상태

12.2 안드로이드 개발 지원 도구



○ [실습 12-3] 복수 에뮬레이터 고급 활용하기

(6) 호스트 명령 창에서 `netstat -anob` 를 입력한다. 호스트 관점에서는 2 개의 에뮬레이터가 동작하고 있으므로, [그림 12-16]처럼 2 개의 에뮬레이터가 동작중인 상태를 확인할 수 있다. adb 클라이언트와 에뮬레이터의

IP 주소가 동일하지만 포트 번호가 다름을 주목하라.

```

C:\> netstat -anob
TCP    127.0.0.1:5037      127.0.0.1:2625      ESTABLISHED  3424
[adb.exe]

TCP    127.0.0.1:5037      127.0.0.1:2641      ESTABLISHED  3424
[adb.exe]

TCP    127.0.0.1:5037      127.0.0.1:2365      ESTABLISHED  3424
[adb.exe]

TCP    127.0.0.1:5554      127.0.0.1:2623      ESTABLISHED  2560
[emulator.exe]

TCP    127.0.0.1:5555      127.0.0.1:2616      ESTABLISHED  2560
[emulator.exe]

TCP    127.0.0.1:5556      127.0.0.1:2358      ESTABLISHED  3000
[emulator.exe]

TCP    127.0.0.1:5557      127.0.0.1:2352      ESTABLISHED  3000
[emulator.exe]
  
```

[그림 12-16] netstat -anob 를 사용한 에뮬레이터 연결 상태 확인

12.2 안드로이드 개발 지원 도구



<저자 한마디> adb 명령을 사용한 파일 이동

명령어 adb pull는

휴대 단말기의 파일이나 에뮬레이터 파일시스템 파일을 호스트 PC로 이동,

명령어 adb push는

호스트 PC의 파일을 주변의 휴대단말기 또는 에뮬레이터 장치로 이동.

adb push <SDK 원본> <에뮬레이터 경로>

adb pull <에뮬레이터 경로> <SDK 경로>



[그림 12-17] adb 를 사용한 파일 이동

12.2 안드로이드 개발 지원 도구



◎ 에뮬레이터

- 가상의 휴대폰 장치
- ARM,
- QEMU 가상머신에
- 안드로이드 커널 탑재
- 윈도우, 리눅스 버전 존재
- Goldfish
- 실제 타깃과 차이: 부동소수점 처리
- `property:ro.kernel.qemu=1`



[그림 12-18] QEMU 가상머신

12.2 안드로이드 개발 지원 도구



◎ [실습 12-4] 에뮬레이터에서 QEMU, /proc 확인하기

- (1) 'adb shell'로 쉘 모드로 간다.
- (2) cd proc로 /proc 디렉토리로 이동한다.
- (3) cat cmdline 명령을 실행한다. cmdline은 커널이 시작될 때 커널에 전달하는 매개변수를 갖는다. qemu=1를 전달하고 있다.

A screenshot of a terminal window titled "Command Shell - adb shell". The terminal shows the following sequence of commands and their outputs:

```
# pwd
pwd
/proc
# cat cmdline
cat cmdline
qemu=1 console=ttyS0 android.checkjni=1 android.gemud=ttyS1 android.ndns=1
#
```

[그림 12-19] QEMU 동작 확인

12.2 안드로이드 개발 지원 도구



◎ [실습 12-4] 에뮬레이터에서 QEMU, /proc 확인하기

(4) `cd /proc/1` 명령을 실행한다. 주 메모리에 생성되는 `/proc` 디렉토리에는 현재 동작중인 프로세스와 커널의 실행상태를 기록하고 있다. `/proc/1`는 부팅시 가장 먼저 시작하는 프로세스 번호이다. `cmdline`을 사용하여 `/proc/1`번 프로세스에 대한 커널로의 입력 변수를 확인해보니 `init`인 것으로 확인된다.

```
# pwd
pwd
/proc/1
# cat cmdline
cat cmdline
/init #
```

[그림 12-20] 프로세스 1번 /proc 확인

12.2 안드로이드 개발 지원 도구



◎ sqlite

- 관계형 데이터베이스
- SQL 표준 명령어 사용
- MySQL 보다 최소 2배 이상 빠르다.
- 기존 임베디드는 JFFS 같은 파일시스템에 데이터 저장
- 안드로이드는 sqlite 사용하여 빠른 검색 특징
- 개별 애플리케이션 마다 /data/data/<패키지이름>/databases/ 생성
- 에뮬레이터 내의 데이터베이스 접속 가능

```
$ adb -s emulator-5554 shell
# sqlite3 /data/data/com.example.google.rss.rssexample/databases/rssitems.db
```

```
CREATE TABLE <테이블 이름> (Column 정의, [테이블제약]);
CREATE TABLE contact_list (id INTEGER, name TEXT, tel TEXT);
```

12.2 안드로이드 개발 지원 도구



◎ [실습 12-5] Sqlite 사용법 및 데이터베이스 읽기

- (1) 데이터베이스 관리 실습을 위하여 에뮬레이터 5554번 1개만을 남겨둔다.
adb shell 쉘모드로 이동한다.
- (2) cd /data/data/com.android.browser/databases 디렉토리로 이동한다.
- (3) ls 명령으로 webview.db 존재를 확인한다.
- (4) sqlite3 명령으로 webview.db 파일을 열어본다.

A screenshot of a terminal window titled "Command Shell - adb shell". The terminal shows the following sequence of commands and outputs:

```
# pwd
pwd
/data/data/com.android.browser/databases
# sqlite3 /data/data/com.android.browser/databases/webview.db
sqlite3 /data/data/com.android.browser/databases/webview.db
SQLite version 3.5.9
Enter ".help" for instructions
sqlite>
```

[그림 12-21] sqlite3 사용법

12.2 안드로이드 개발 지원 도구



○ [실습 12-5] Sqlite 사용법 및 데이터베이스 읽기

(5) /data/system/ 디렉토리에 있는 syncmanager.db 데이터베이스 내용을 살펴본다. sqlite를 사용하여 [그림 12-22]처럼 데이터베이스에 접속을 하고, '.tables' 명령을 입력하면 active, history, settings, status 등의 테이블이 나온다.

(6) select * from history 명령을 입력한다. history 테이블의 내용을 본다.

```

C:\ Command Shell - adb shell
# sqlite3 syncmanager.db
sqlite3 syncmanager.db
SQLite version 3.5.9
Enter ".help" for instructions
sqlite> .tables
.tables
active          history          settings         status
android_metadata pending          stats
sqlite> select * from history;
select * from history;
15|1|1270865469242|13451|2|1|0|0|success
16|2|1270865483379|28700|2|1|0|0|success
17|3|1270865512849|13165|2|1|0|0|success
18|4|1270865526336|2592|2|1|0|0|success
19|3|1270865529187|222|1|1|0|0|success
20|2|1270865529573|1252|0|1|0|0|success
  
```

[그림 12-22] 데이터베이스 내용보기

12.2 안드로이드 개발 지원 도구



- ◎ 기타도구
- ◎ Dx
- ◎ Dexdump
- ◎ Ant
- ◎ Valgrind
- ◎ AIDL

12.3 안드로이드용 busybox 설치



- 리눅스 1,000 여 개의 시스템 관리를 위한 명령어: toolbox 제공
- 안드로이드 toolbox는 단지 50 여개의 리눅스 명령어 만 제공

```

Command Shell - adb shell
-rwxr-xr-x root shell 44536 2009-07-01 20:41 dhcpcd
-rwxr-xr-x root shell 18892 2009-07-01 20:41 hciattach
-rwxr-xr-x root shell 10688 2009-07-01 20:41 showlease
lrwxr-xr-x root shell 2009-07-02 15:23 rmod -> toolbox
lrwxr-xr-x root shell 2009-07-02 15:23 umount -> toolbox
lrwxr-xr-x root shell 2009-07-02 15:23 dmesg -> toolbox
-rwxr-xr-x root shell 86944 2009-07-01 20:41 sh
lrwxr-xr-x root shell 2009-07-02 15:23 kill -> toolbox
lrwxr-xr-x root shell 2009-07-02 15:23 rm -> toolbox
-rwxr-xr-x root shell 9764 2009-07-01 20:41 service
lrwxr-xr-x root shell 2009-07-02 15:23 chmod -> toolbox
-rwxr-xr-x root shell 189 2009-07-01 20:41 an
-rwxr-xr-x root shell 14024 2009-07-01 20:41 installd
lrwxr-xr-x root shell 2009-07-02 15:23 ls -> toolbox
-rwxr-xr-x root shell 5420 2009-07-01 20:41 dalvikvm
lrwxr-xr-x root shell 2009-07-02 15:23 cmp -> toolbox
-rwxr-xr-x root shell 9828 2009-07-01 20:41 servicemanager
lrwxr-xr-x root shell 2009-07-02 15:23 sendevent -> toolbox
-rwxr-xr-x root shell 151868 2009-07-01 20:41 gdbserver
lrwxr-xr-x root shell 2009-07-02 15:23 start -> toolbox
-rwxr-xr-x root shell 72604 2009-07-01 20:41 toolbox
lrwxr-xr-x root shell 2009-07-02 15:23 route -> toolbox
-rwxr-xr-x root shell 5496 2009-07-01 20:41 gzip
lrwxr-xr-x root shell 2009-07-02 15:23 smd -> toolbox
lrwxr-xr-x root shell 2009-07-02 15:23 netstat -> toolbox
-rwxr-xr-x root shell 149444 2009-07-01 20:41 akmd
-rwxr-xr-x root shell 194 2009-07-01 20:41 ine
lrwxr-xr-x root shell 2009-07-02 15:23 bugreport -> dumpstat
  
```

[그림 12-23] toolbox에 들어있는 명령어들

12.3 안드로이드용 busybox 설치



- ◎ 단점 : cp, tar 같은 명령어 없어 개발 및 디버그시 불편
- ◎ busybox를 통한 해결

Busybox 소스 다운로드하기

12.3 안드로이드용 busybox 설치



◎ [실습 12-6] busybox 설치하기

- (1) 명령창을 연다.
- (2) d: 로 하드디스크 D로 이동한다.
- (3) cd D:\Wandroid\Wandroid-sdk\tools 명령을 실행한다.
- (4) adb push busybox /data 를 입력한다. adb를 이용해 SDK의 busybox 실행파일을 에뮬레이터 파일시스템 /data 아래로 복사한다.
- (5) adb shell 명령어를 입력한다. adb를 이용해 android 에뮬레이터의 shell(/system/bin/sh) 명령 실행 후 쉘 프롬프트 # 로 전환한다.
- (6) cd data; ls -l 명령으로 복사된 busybox 파일 존재 여부를 확인한다. [그림 12-25]처럼 busybox가 설치된 것을 확인한다.
- (7) chmod 777 busybox 명령을 실행하여 busybox 명령을 사용하기 위해 실행 가능한 권한으로 설정한다.

**[그림 12-25] busybox 파일의
에뮬레이터 설치**

```

Command Shell - adb shell
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Program Files\Windows Resource Kits\Tools>d:

D:\>cd D:\Wandroid\Wandroid-sdk\tools

D:\Wandroid\Wandroid-sdk\tools>adb push busybox /data
675 KB/s < 0 bytes in 2236784.003s>

D:\Wandroid\Wandroid-sdk\tools>adb shell
# cd data
cd data
# ls -l
ls -l
-rwxr-xr-x root root 2236784 2009-04-16 02:47 busybox
drwxr-xr-x system misc 2010-06-02 02:52 misc
drwxr-xr-x shell shell 2010-06-02 02:52 local
drwxr-xr-x system system 2010-06-02 02:52 app-private
drwxr-xr-x system system 2010-06-02 02:54 backup
drwxr-xr-x root root 2010-06-02 02:54 property
drwxr-xr-x system system 2010-06-06 05:37 anr
drwxr-xr-x system system 2010-06-06 07:38 data
drwxr-xr-x root log 2010-06-02 02:52 dontpanic
drwxr-xr-x system system 2010-06-13 03:46 dalvik-cache
drwxr-xr-x system system 2010-06-13 03:46 app
drwxr-xr-x system system 2010-06-13 03:46 system
drwxr-xr-x root root 2010-06-02 02:52 lost-found
#
  
```



12.4 파일 추출 실습

◎ [실습 12-7] 파일 압축 및 추출하기

(1) adb shell 쉘모드로 이동한다.

(2) /data/busybox tar cvf /data/system.tar /system을 입력한다. busybox의 tar 명령을 사용하여 에뮬레이터의 /system 디렉토리를 /data/system.tar 파일로 압축한다. [그림 12-26]은 에뮬레이터 /system 디렉토리의 압축 및 복사하기를 보여준다. [그림 12-27]은 에뮬레이터 /data 디렉토리에 system.tar 압축 파일이 복사된 것을 ls -l 명령어로 보여준다.

```

Command Shell - adb shell
# pwd
pwd
/
# cd data
cd data
# /data/busybox tar cvf /data/system.tar /system
/data/busybox tar cvf /data/system.tar /system
tar: removing leading '/' from member names
system/
system/app/
system/app/Browser.apk
system/app/EnhancedGoogleSearchProvider.apk
system/app/Music.apk
system/app/MediaProvider.apk
system/app/NetSpeed.apk
system/app/Calculator.apk
  
```

[그림 12-26] 에뮬레이터 /system 디렉토리의 압축 및 복사하기

```

Command Shell - adb shell
# pwd
pwd
/data
# ls -l
ls -l
-rw-r--r-- root    root    45321728 2010-06-13 04:43 system.tar
-rwxrwxrwx root    root    2236784 2009-04-16 02:47 busybox
drwxrwx--t system  misc    2010-06-02 02:52 misc
drwxrwx--x shell   shell   2010-06-02 02:52 local
drwxrwx--x system  system  2010-06-02 02:52 app-private
drwx----- system  system  2010-06-02 02:54 backup
drwx----- root    root    2010-06-02 02:54 property
drwxrwx--x system  system  2010-06-06 05:37 anr
drwxrwx--x system  system  2010-06-06 07:38 data
drwxr-x--- root    log     2010-06-02 02:52 dontpanic
drwxrwx--x system  system  2010-06-13 03:46 dalvik-cache
drwxrwx--x system  system  2010-06-13 03:46 app
drwxrwx--x system  system  2010-06-13 04:45 system
drwxrwx--- root    root    2010-06-02 02:52 lost+found
#
  
```

[그림 12-27] system.tar 복사확인

12.4 파일 추출 실습



◎ [실습 12-7] 파일 압축 및 추출하기

(3) exit 명령어 수행하여 shell을 빠져나온다.

(4) 에뮬레이터 /data/ 디렉토리에 있는 system.tar 압축 파일을 데스크탑 컴퓨터의 SDK로 추출하는 실습이다. adb pull /data/system.tar system.tar 명령을 실행한다. [그림 12-28]는 system.tar 가 SDK의 /tools 디렉토리에 복사된 것을 보여준다.

```

Command Shell
D:\android\android-sdk\tools>adb pull /data/system.tar system.tar
819 KB/s <0 bytes in 45321728.053s>

D:\android\android-sdk\tools>dir
D 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 449B-724C

D:\android\android-sdk\tools 디렉터리

2010-06-13 오후 02:31 <DIR>      .
2010-06-13 오후 02:31 <DIR>      ..
2010-06-02 오전 11:17          578,611 adb.exe
2010-06-02 오전 11:17          96,256 adbwinapi.dll
2010-06-02 오전 11:17          60,928 adbwinusbapi.dll
2010-06-02 오전 11:17           3,394 android.bat
2010-06-02 오전 11:17           1,565 apkbuilder.bat
2009-04-16 오전 11:47       2,236,784 busybox
2010-06-02 오전 11:17           2,129 ddms.bat
2010-06-02 오전 11:17        302,823 dntracedump.exe
2010-06-02 오전 11:17           1,491 draw9patch.bat
2010-06-02 오전 11:17       8,596,234 emulator.exe
2010-06-02 오전 11:17          17,976 emulator_NOTICE.txt
2010-06-02 오전 11:17          921,075 etctest.exe
2010-06-02 오전 11:17          336,613 fastboot.exe
2010-06-02 오전 11:17           1,528 hierarchyviewer.bat
2010-06-02 오전 11:17          216,987 hprof-conv.exe
2010-06-02 오전 11:17 <DIR>      Jet
2010-06-02 오전 11:17           1,680 layoutopt.bat
2010-06-02 오전 11:17 <DIR>      lib
2010-06-02 오전 11:17          212,741 nkscard.exe
2010-06-02 오전 11:17          205,851 NOTICE.txt
2010-06-02 오전 11:17           311 source.properties
2010-06-02 오전 11:17       1,627,623 sqlite3.exe
2010-06-13 오후 02:32       45,321,728 system.tar
2010-06-02 오전 11:17           1,982 traceview.bat
2010-06-02 오전 11:17          628,727 zipalign.exe
23개 파일          61,375,037 바이트
4개 디렉터리       9,933,852,672 바이트 남음
  
```

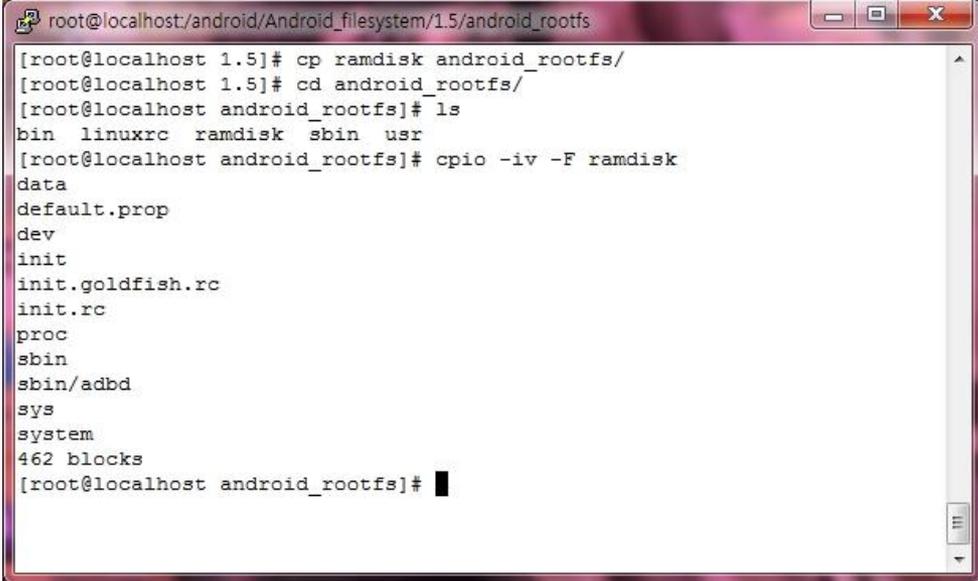
[그림 12-28] SDK 디렉토리로
파일 추출결과



12.4 파일 추출 실습

◎ cpio 사용

- 이미지 압축파일 관리 도구
- ramdisk, sdcard 이미지 파일 관리

A terminal window with a title bar showing the path 'root@localhost/android/Android_filesystem/1.5/android_rootfs'. The terminal contains the following text:

```
[root@localhost 1.5]# cp ramdisk android_rootfs/
[root@localhost 1.5]# cd android_rootfs/
[root@localhost android_rootfs]# ls
bin linuxrc ramdisk sbin usr
[root@localhost android_rootfs]# cpio -iv -F ramdisk
data
default.prop
dev
init
init.goldfish.rc
init.rc
proc
sbin
sbin/adbd
sys
system
462 blocks
[root@localhost android_rootfs]#
```

[그림 12-29] 램디스크 이미지 내용 확인

12.5 에뮬레이터에 SD를 가상장치로 연결

- ◎ SD는 대용량 메모리 장치
- ◎ 플래시 메모리 32MB 로 동영상 파일 저장에 한계
- ◎ 주변장치로 활용
- ◎ SD : Secure digital
- ◎ SD는 골드피시 장치에 속하지 않음
- ◎ 따라서 에뮬레이터에 SD를 추가하는 방법 필요

12.5 에뮬레이터에 SD를 가상장치로 연결

○ [실습 12-8] SD 카드 생성 및 파일 추출하기

- (1) 명령창을 띄운다.
- (2) D:\Wandroid\Wandroid-sdk\tools 디렉토리로 이동한다.
- (3) mkcard 256M sd.img 명령을 실행한다.
- (4) dir 명령을 실행하여 [그림 12-30]처럼 256MB의 sd.img가 생성된 것을 확인한다.

```

C:\> Command Shell
D:\Wandroid\Wandroid-sdk\tools>mkcard 256M sd.img

D:\Wandroid\Wandroid-sdk\tools>dir
D 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 449B-724C

D:\Wandroid\Wandroid-sdk\tools 디렉터리

2010-06-13 오후 03:06 <DIR>      .
2010-06-13 오후 03:06 <DIR>      ..
2010-06-02 오후 11:17             578,611 adb.exe
2010-06-02 오후 11:17             96,256 adbWinApi.dll
2010-06-02 오후 11:17             60,928 adbWinUsbApi.dll
2010-06-02 오후 11:17             3,394 android.bat
2010-06-02 오후 11:17             1,565 apkbuilder.bat
2009-04-16 오후 11:47        2,236,784 busybox
2010-06-02 오후 11:17             2,129 ddms.bat
2010-06-02 오후 11:17        302,823 dmtracedump.exe
2010-06-02 오후 11:17             1,491 draw9patch.bat
2010-06-02 오후 11:17        8,596,234 emulator.exe
2010-06-02 오후 11:17             17,976 emulator_NOTICE.txt
2010-06-02 오후 11:17            921,075 etcitool.exe
2010-06-02 오후 11:17            336,613 fastboot.exe
2010-06-02 오후 11:17             1,528 hierarchyviewer.bat
2010-06-02 오후 11:17            216,987 hprof-conv.exe
2010-06-02 오후 11:17 <DIR>      Jet
2010-06-02 오후 11:17             1,680 layoutopt.bat
2010-06-02 오후 11:17 <DIR>      lib
2010-06-02 오후 11:17            212,741 mkcard.exe
2010-06-02 오후 11:17            205,851 NOTICE.txt
2010-06-13 오후 03:06        268,435,456 sd.img
2010-06-02 오후 11:17             311 source.properties
  
```

[그림 12-30] sd.img 파일 생성 확인

12.5 에뮬레이터에 SD를 가상장치로 연결

◎ [실습 12-8] SD 카드 생성 및 파일 추출하기

- (5) sd.img를 c:\temp로 복사한다. \tools 폴더에 sd.img를 넣고 SD 카드 실습하면 에뮬레이터에 가상장치로 연결이 되지 않고 있다. 그래서 다른 폴더에 옮겨 놓는 것이다.
- (6) emulator -avd myAVD -sdcard c:\temp\sd.img 명령을 실행하여 에뮬레이터에 sd.img 이미지를 가상장치로 연결한다.



```
Command Shell - emulator -avd myAVD -sdcard c:\temp\sd.img
C:\Program Files\Windows Resource Kits\Tools>emulator -avd myAVD -sdcard c:\temp\sd.img
```

[그림 12-31] AVD 지원에 의한 sdcard 장치 연결 명령

12.5 에뮬레이터에 SD를 가상장치로 연결

◎ [실습 12-8] SD 카드 생성 및 파일 추출하기

- (7) 에뮬레이터에서 setting 메뉴 -> SD card & phone storage settings를 클릭한다. [그림 12-31] 실행결과 자동으로 에뮬레이터가 구동되고 SD 카드가 인식된 것을 [그림 12-32] 1)과 2)처럼 확인할 수 있다. 약 256MB 메모리 용량이 할당되어 있다. 여기서 avd 이름은 이클립스 'Run Configurations' 메뉴에서 target 탭의 avd 이름 설정에서 사용한 것과 동일하여야 한다. 에뮬레이터에는 'myAVD'라는 avd가 가상장치로서 연결되었고, 이제 sd.img 이미지가 SD 카드로서 연결이 되어 사용가능하다.



1) 초기화면 2) SD 가상장치 연결
[그림 12-32] SD 카드 인식화면

12.5 에뮬레이터에 SD를 가상장치로 연결

◎ [실습 12-8] SD 카드 생성 및 파일 추출하기

(8) SDK 에 있는 busybox 파일을 에뮬레이터 /sdcard 디렉토리에 복사 과정을 실습한다. [그림 12-31]의 명령창이 정지해 있음으로 또 하나의 명령창을 열어서 SDK 에 있는 busybox 파일을 에뮬레이터 /sdcard 디렉토리에 복사하는 실습을 한다. [그림 12-33]에 나타난 것처럼 adb push busybox /sdcard 명령을 실행한다.



```
Command Shell - adb shell

D:\android\android-sdk\tools>adb push busybox /sdcard
621 KB/s <0 bytes in 2236784.003s>

D:\android\android-sdk\tools>adb shell
# cd /sdcard
cd /sdcard
# ls -l
ls -l
d---rwxr-x system    sdcard_rw          2010-06-13 07:11 LOST.DIR
----rwxr-x system    sdcard_rw  2236784 2009-04-16 02:47 busybox
#
```

[그림 12-33] 또 다른 명령창 열기 및 busybox 추출하기

12.5 에뮬레이터에 SD를 가상장치로 연결

◎ [실습 12-8] SD 카드 생성 및 파일 추출하기

(9) 에뮬레이터 가상장치인 SD 카드에 있는 파일을 SDK로 복사하는 과정이다. `adb pull /data/app/com.corea.Android.apk .` 명령을 실행한다. /data/app 디렉토리에 있는 `com.corea.Android.apk` 파일을 데스크탑

SDK의 하드디스크로 다운로드 하는 과정을 [그림 12-34]에 보여준다.

```

C:\> Command Shell
D:\android\android-sdk\tools>adb pull /data/app/com.corea.Android.apk .
831 KB/s (0 bytes in 13299.000s)

D:\android\android-sdk\tools>dir
D 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 449B-724C

D:\android\android-sdk\tools 디렉터리

2010-06-13 오후 04:51 <DIR>      .
2010-06-13 오후 04:51 <DIR>      ..
2010-06-02 오전 11:17          578,611 adb.exe
2010-06-02 오전 11:17          96,256 AdbWinApi.dll
2010-06-02 오전 11:17          60,928 AdbWinUsbApi.dll
2010-06-02 오전 11:17           3,394 android.bat
2010-06-02 오전 11:17           1,565 apkbuilder.bat
2009-04-16 오전 11:47       2,236,784 busybox
2010-06-13 오후 04:51          13,299 com.corea.Android.apk
2010-06-02 오전 11:17           2,129 ddms.bat
2010-06-02 오전 11:17       302,823 dntracedump.exe
2010-06-02 오전 11:17           1,491 draw9patch.bat
2010-06-02 오전 11:17       8,596,234 emulator.exe
2010-06-02 오전 11:17          17,976 emulator_NOTICE.txt
2010-06-02 오전 11:17          921,075 etc1tool.exe
2010-06-02 오전 11:17         336,613 fastboot.exe
2010-06-02 오전 11:17           1,528 hierarchyviewer.bat
2010-06-02 오전 11:17          216,987 hprof-conv.exe
  
```

[그림 12-34] com.corea.Android.apk 파일 추출결과

12.5 에뮬레이터에 SD를 가상장치로 연결

◎ [실습 12-8] SD 카드 생성 및 파일 추출하기

(10) 리눅스에서 실제 타겟을 사용한 실습 과정이다. 본 실습을 위해 하이버스의 xhyper320tku 보드를 사용하여야 하며 구체적인 것은 연습문제를 참고한다. 에뮬레이터의 SD 카드에 들어 있는 apk 파일을 실행하기 위해서는 안드로이드 애플리케이션 소스에서 변경이 필요하다. 통상적인 애플리케이션 apk는 개발시 /data/data/packagename/에 생성 및 저장된다. /data/data/packagename/가 아닌 SD 카드에 저장된 애플리케이션 파일을 실행하려면 자바소스에서 데이터에 대한 경로 설정을 변경해 주어야 한다. 예를 들면, 비디오를 재생하려면 자바 소스에서 SD 카드의 데이터를 사용할 수 있도록 Video 클래스의 해당 메소드를 변경해 주어야 한다. 즉, Video.java 프로그램 소스에서 Video.setVideoPath("/sdcard/MiniRobot.mp4")를 수정한다. 그러면 애플리케이션에서 비디오 데이터를 /sdcard에서 읽게 된다. 본 실습에 대한 구체적인 것은 연습문제를 참고한다.



끝 !