



안드로이드 기초

학습 목표 및 내용

■ 학습목표

- 모바일 운영체제의 일반적인 특징과 안드로이드의 주요 특징 및 전체적인 아키텍처를 알아본다.
- 안드로이드 학습을 위한 개발 환경을 설치하고 원활한 실습을 위한 환경 설정 방법을 알아본다.

스마트폰

■ 모바일 환경

- 초기의 스마트 폰 : 속도, 용량 부족. 네트워크 비연결
- 아이폰 이후 대중화되고 모바일 소프트웨어의 수요도 폭발적으로 증가하였다.
- 피쳐폰과의 차이점 : 소프트웨어 추가 설치 가능하여 용도의 제한이 없다.



스마트폰

- 불특정 다수의 소프트웨어를 중앙에서 통제하는 운영체제가 필요하다. 자원을 분배 및 관리하며 공통 API를 제공한다.
- 전화기 운영체제
 - 모바일 운영체제
 - 모바일에서 실행되는 운영체제.
 - 모바일 하드웨어 자원을 직접 제어하고 관리하는 시스템 소프트웨어.
 - 데스크톱 운영체제에 비해 소규모

스마트폰

■ 모바일 운영체제

● 피쳐폰의 특징

- 별도의 운영체제가 없고 통화 기능만 제공.
- 제조사에서 미리 설치해 놓은 프로그램만 사용 가능.
- 제한적인 기능만 제공.
- 소프트웨어 추가 설치가 거의 불가.

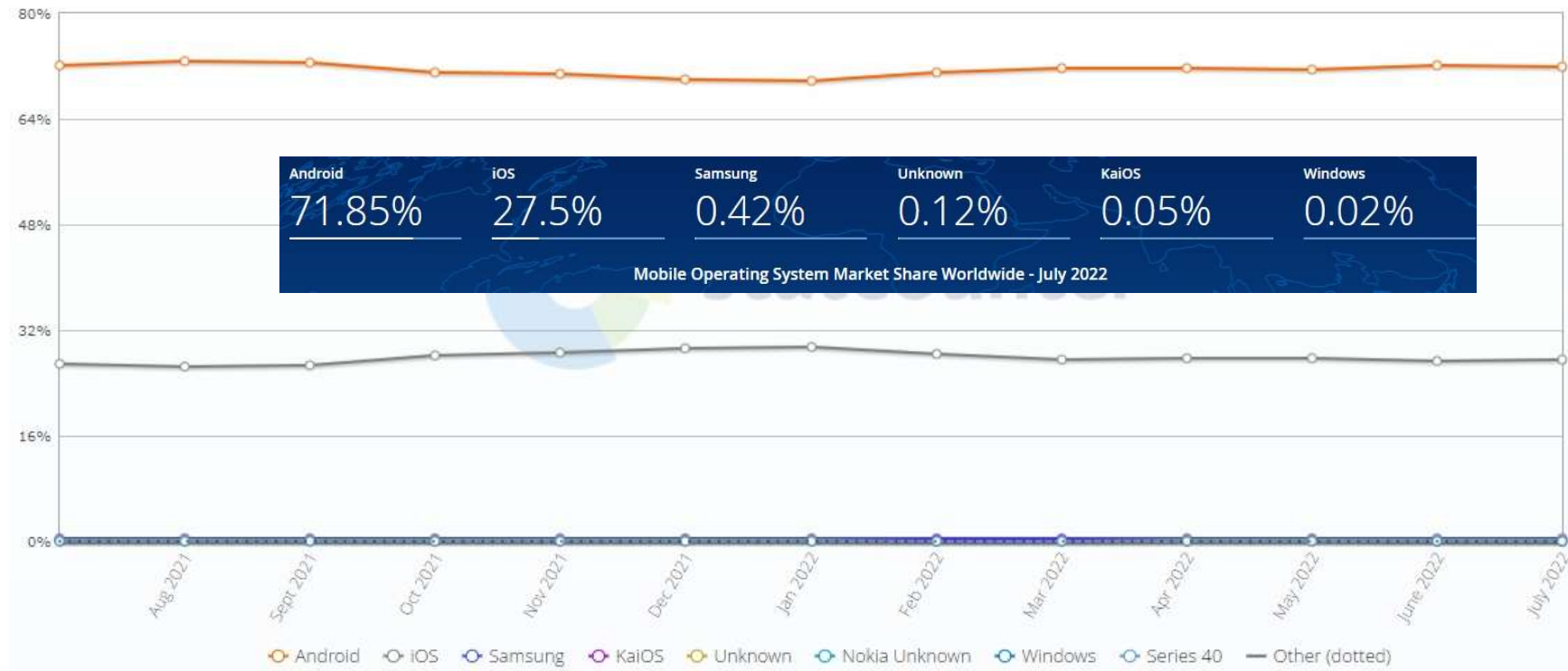
● 스마트폰의 특징

- 필요에 따라 프로그램의 추가 설치 가능.
- 폰에 별도의 운영체제가 필요.
- 다양한 기능 제공.
- 소프트웨어 추가 설치가 가능.

스마트폰



■ 모바일 운영체제 점유율



- 출처: <https://gs.statcounter.com/os-market-share/mobile/worldwide>

안드로이드

■ 등장

- 개발 주체는 OHA(Open Handset Alliance) 컨소시엄
- 실질적인 리더는 인터넷 검색 업체인 구글(Google)



안드로이드

■ 특징

- 비독점 개방 플랫폼을 지향
- 개방 플랫폼 vs 비개방 플랫폼

[개방 플랫폼]

- 라이선스 비용이 없음.
- 개발이 자유로움.
- 개발툴에 문제 발생 시, 기술 지원을 받을 수 없음.

[비개방 플랫폼]

- 라이선스 비용 지불.
- 개발 환경이 잘 구축되어 있음.
- 문제 발생 시, 기술 지원이 가능.

안드로이드

■ 특징

- 운영체제의 핵심이라고 할 수 있는 커널은 공개 운영체제인 리눅스 기반
- 안드로이드 개발에 공식적으로 자바 언어(최근 Kotlin도 포함)를 사용
- 검증된 많은 라이브러리들을 대거 포함하고 있어 웬만한 기능은 별도의 외부 라이브러리를 사용할 필요 없음
- 플랫폼에 내장된 빌트인 프로그램과 사용자가 만든 프로그램이 동일한 API를 사용하므로 모든 프로그램은 평등함
- 개방된 환경인만큼 개발툴과 관련 문서들이 모두 무료로 제공
- 초기 안드로이드의 단점
 - 오래되지 않은 만큼 숙련된 개발자가 많지 않으며, 관련 자료도 부족함
 - 공식 문서가 아직 완비되지 않은 상태
 - SDK도 너무 자주 업그레이드되어 안정적이지 못하며 개발툴의 사소한 버그들도 상당히 많은 편
 - 파편화(Fragmentation) : 개방된 환경 하에 통제 불가능한 변종의 출현
 - 명확한 지원을 기대할 수 없음

안드로이드

■ 안드로이드와 자바

- 현재 자바의 최신 버전은 Java 16, 안드로이드는 아직 Java 8까지만 지원
 - 람다식 및 메소드 참조
 - 인터페이스의 디폴트 및 정적 메소드

■ 안드로이드 앱 개발 언어

- iOS는 Objective-C, Swift
- 안드로이드는 Java, Kotlin

안드로이드

■ 아키텍처

- 효율적이고 안정적인 자원 관리를 위해 계층 구성
- 하위에서는 저수준 지원 요소, 상위로 올라갈수록 응용 요소
- 추상화와 보안을 위한 인접한 계층끼리만 통신 가능
- 개발자의 주 학습 대상은 프레임워크이다.



안드로이드

■ 아키텍처

① 리눅스 커널

- 백그라운드 스레드, 메모리 관리, 보안, 디바이스 드라이버 등의 저수준 관리 기능을 담당
- 디바이스 드라이버의 지원이 광범위함.

② 라이브러리

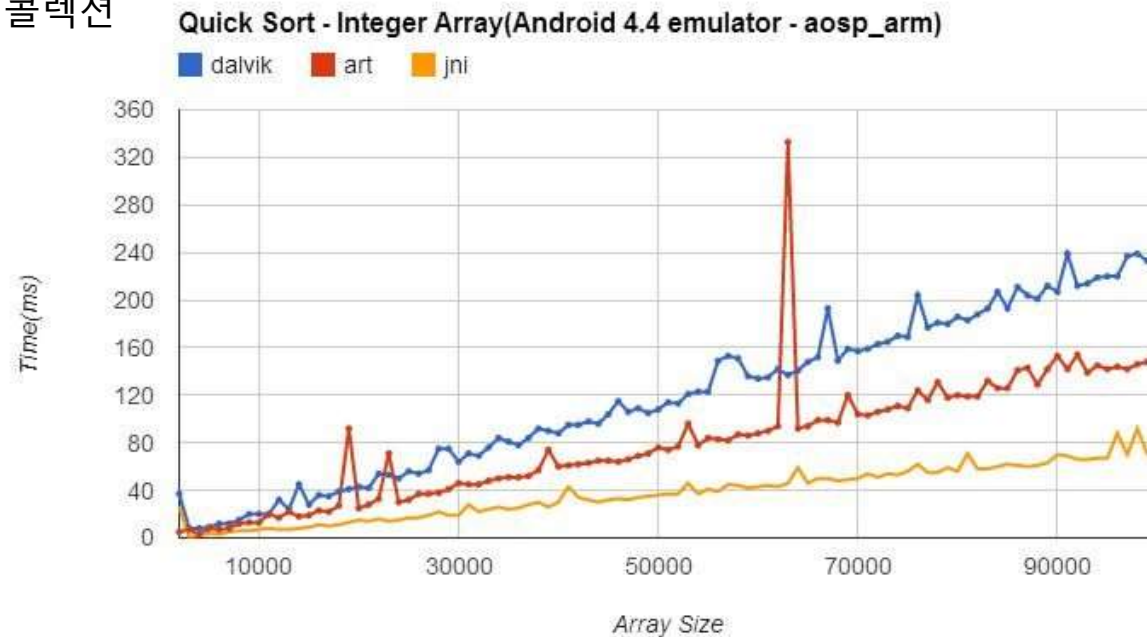
- 라이브러리는 장비의 전반적인 속도를 결정하는 중요한 요소임.
- 공식 언어인 자바가 아니라 C로 작성되어 있음.
- 안드로이드 라이브러리 종류
 - BSD 파생 libc 라이브러리
 - 2D 그래픽을 위한 서피스 관리자 및 SGL, - 3D 그래픽을 위한 OpenGL
 - 벡터 폰트 출력을 위한 FreeType - 오디오, 비디오 재생을 위한 미디어 지원
 - 데이터베이스 지원을 위한 SQLite
 - 인터넷 브라우징을 위한 SSL과 크롬, 사파리에 채용된 WebKit

안드로이드

■ 아키텍처

③ 런타임

- 달빅 가상 머신과 자바 코어 라이브러리로 구성
- 자바 가상 머신 대신 모바일 환경에 최적화한 달빅(Dalvik) 가상 머신을 사용함.
- 프로세스 단위로 별도의 달빅 가상 머신이 할당되므로 안정성이 높고 메모리 사용량을 줄여 복수 개의 가상 머신도 효율적으로 동작하도록 설계되었음.
- 5.0 이후 컴파일 방식과 가비지 컬렉션을 개선한 ART 런타임 채용
- Ahead-of-time (AOT) 컴파일 :
 - 필요할 때마다 앱을 컴파일하여서 실행하는 달빅과 달리 ART는 미리 앱을 컴파일
 - 향상된 가비지 컬렉션
 - 디버깅 향상



안드로이드

■ 아키텍처

④ 프레임워크

- 안드로이드의 API.
- 응용 프로그램들은 하위의 커널이나 시스템 라이브러리를 직접적으로 호출할 수 없으며 API를 통해서 기능을 요청해야 함.
- API는 하드웨어 접근을 추상화하며 UI 제작에 필요한 위젯을 정의하고 시스템 라이브러리와 통신할 수 있도록 함.

⑤ 응용프로그램

- 인터넷 브라우저, 바탕화면, 주소록 등 플랫폼과 함께 설치되는 애플릿들은 물론이고 마켓에서 다운로드 받아 설치하는 게임, 유틸리티 등도 모두 이 수준에서 실행됨.

안드로이드

■ 안드로이드 실행

- 안드로이드 실행 파일은 전통적인 실행 파일과 다름.
 - 안드로이드의 실행 파일
 - 실행 파일은 같은 패키지에 속한 자바 클래스와 리소스의 집합일 뿐이며 프로세스와 반드시 1 : 1로 대응되지 않음.
 - 응용 프로그램끼리 서로의 기능을 공유할 수 있음.
 - 다른 프로그램의 구성 요소를 불러와 같은 주소 공간에서 실행할 수 있음. (윈도우의 COM이나 OMG의 CORBA 등과 개념적으로 유사함)
 - 안드로이드 응용 프로그램은 적절한 권한만 있으면 누구든지 인스턴스화할 수 있는 4개의 컴포넌트로 구성됨.
 - main 같은 유일한 진입점이 따로 없으며 처음으로 생성되는 인스턴스의 생성자가 실질적인 진입점이 됨
- 전통적 실행 파일의 경우
 - 실행 파일에는 프로그램을 구성하는 코드와 데이터가 들어 있음.
 - 실행 파일이 메모리상으로 올라와 실행을 시작하면 프로세스가 됨.
 - 실행 파일과 프로세스가 1:1로 대응되며 응용 프로그램끼리는 서로 독립적임.

안드로이드

■ 안드로이드 실행

- 인스턴화할 수 있는 4개의 컴포넌트 : 액티비티, 서비스, 방송 수신자, 콘텐츠 제공자

[액티비티]

- 사용자 인터페이스를 구성하는 기본 단위다. 눈에 보이는 화면 하나가 액티비티이며 여러 개의 뷰들로 구성
- 응용 프로그램은 필요한 만큼의 액티비티를 가질 수 있으며 그 중 어떤 것을 먼저 띄울지를 지정

[서비스]

- UI가 없어 사용자 눈에 직접적으로 보이지 않으며 백그라운드에서 무한히 실행되는 컴포넌트
- 전형적인 예로 미디어 플레이어를 들 수 있는데 비활성화된 상태라도 노래는 계속 재생
- UI가 없으므로 사용자의 명령을 받아들일 수 있는 액티비티와 연결해서 사용됨

안드로이드

■ 안드로이드 실행

- 인스턴화할 수 있는 4개의 컴포넌트 : 액티비티, 서비스, 방송 수신자, 콘텐츠 제공자

[방송수신자(Broadcast Receiver)]

- 시스템으로부터 전달되는 방송을 대기하고 신호 전달 시 수신하는 역할
- 예를 들어 배터리가 떨어졌다거나 사진을 찍었다거나 네트워크 전송이 완료 되었다는 등의 신호를 받음
- 신호만 대기할 뿐 UI를 따로 가지지는 않으므로 방송 수신 시 방송의 의미를 해석하고 적절한 액티비티를 띄우는 역할

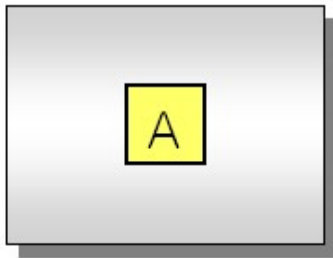
[콘텐츠 제공자(Content Provider)]

- 다른 응용 프로그램을 위해 자신의 데이터를 제공
- 안드로이드는 보안이 엄격하여 다른 응용 프로그램의 데이터를 함부로 액세스 하지 못함
- 응용 프로그램 간에 데이터를 공유할 수 있는 합법적인 유일한 장치가 바로 콘텐츠 제공자

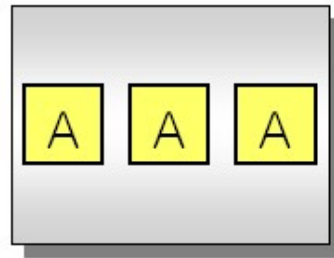
안드로이드

■ 안드로이드 실행

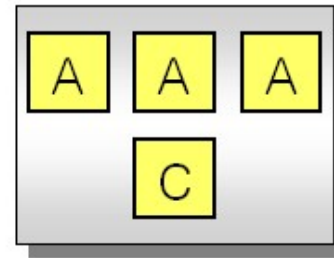
- 응용 프로그램은 액티비티, 서비스, 방송 수신자, 콘텐츠 제공자 중 일부만을 가질 수도 있고 여러 개를 가질 수도 있음



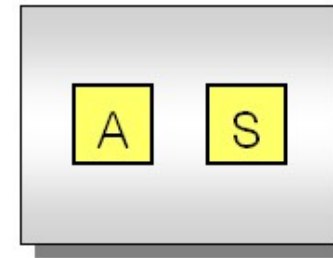
간단한 예제



보기, 추가, 편집 등의 기능을 가지는 주소록



공유 가능한 주소록



MP3 플레이어

안드로이드 역사 및 버전별 특징



■ 역사

● 개발

- 2005년 7월, 개발 시작
- 2007년 11월, 안드로이드 첫 버전 릴리즈
- 2008년 10월, 최초의 상용 안드로이드 폰 출시(HTC사의 G1)
- 현재 점유율 1위

■ 버전 별 특징

- https://ko.wikipedia.org/wiki/%EC%95%88%EB%93%9C%EB%A1%9C%EC%9D%B4%EB%93%9C_%EB%B2%84%EC%A0%84_%EC%97%AD%EC%82%AC

■ 안드로이드 코드 이름

- 초기에는 Cupcake, Donut, Éclair, Froyo 등과 같은 디저트 이름을 사용
- 2019년 '안드로이드10'을 내놓으면서 정식 명칭을 숫자로만 표기
- 그러나 내부적으로는 여전히 디저트 이름을 사용