

모바일 프로그래밍

11 프로세스와 스레드 1

2017 2학기

강승우

프로세스와 스레드 개념

프로세스(Process)와 스레드(Thread)

- 프로세스란?
- 스레드란?
- Multi-threading?

프로세스와 스레드

- 프로세스

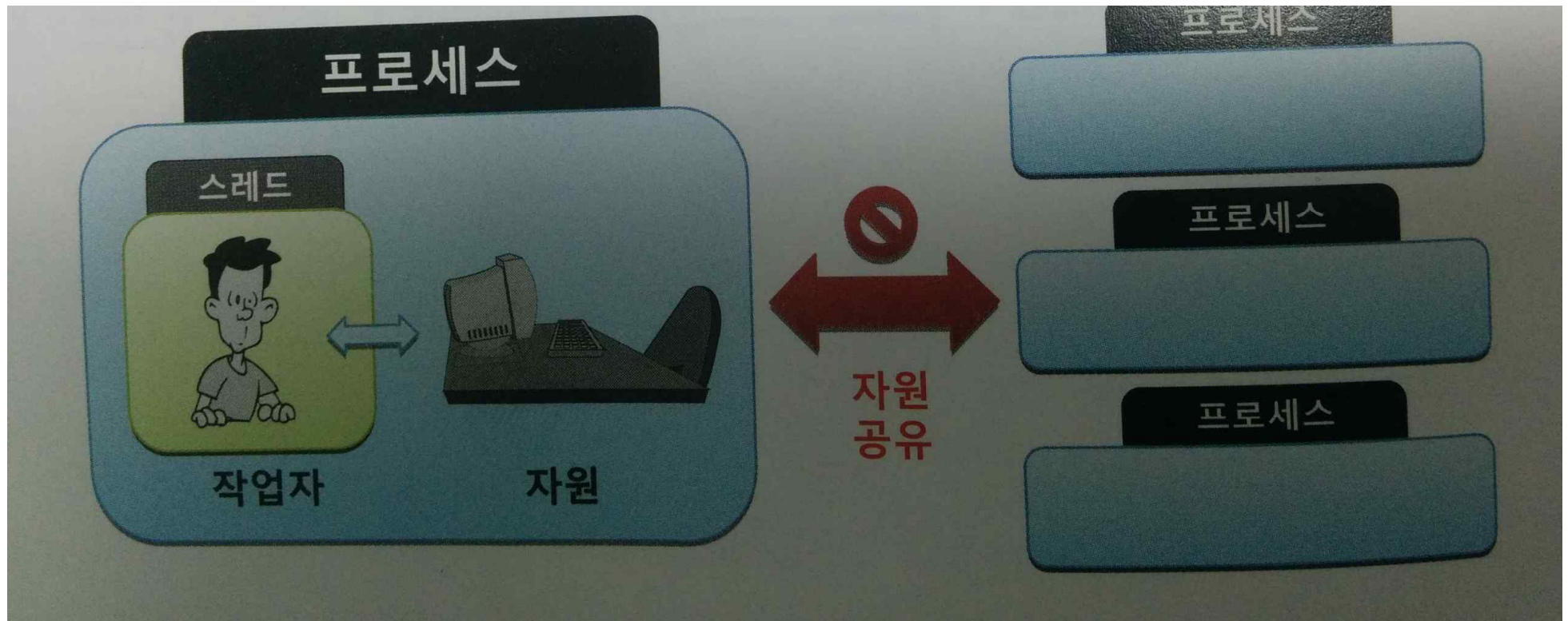
- 실행 중인 프로그램과 그것의 상태를 나타내는 단위
- 프로그램 실행에 필요한 자원이 할당됨
 - 메모리 주소 공간, 파일 디스크립터, ..
- 하나 이상의 스레드를 가질 수 있음
- 프로세스 간에는 자원이 공유되지 않음

- 스레드

- 프로그램의 실행 흐름을 표현하는 개념
- 프로그램 실행을 담당하는 작업자라고 볼 수 있음
- 한 프로세스에 속한 다른 스레드와 프로세스의 자원을 공유함
- 멀티스레드 프로그램에서 동기화 문제 발생 가능

프로세스와 스레드의 비유

- 회사: 프로세스
- 회사 내 서류, 책상, 의자, 컴퓨터: 자원
- 자원을 가지고 일하는 직원: 작업자 (스레드)



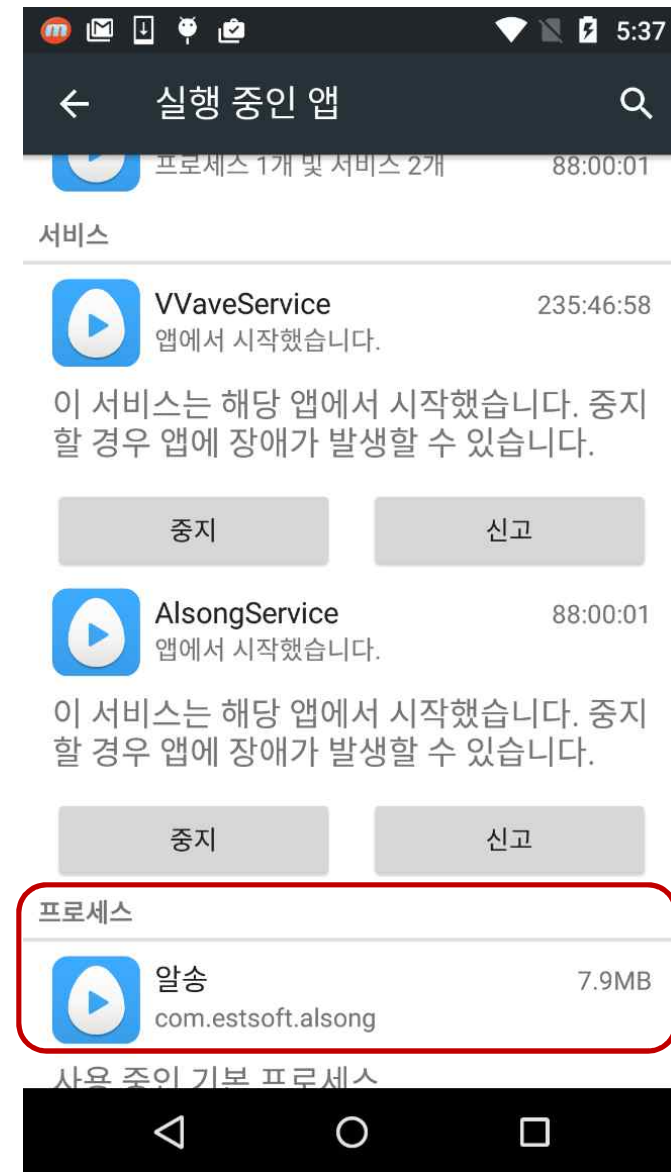
Multi-thread를 통한 다중 처리




안드로이드의 프로세스와 스레드

- 안드로이드

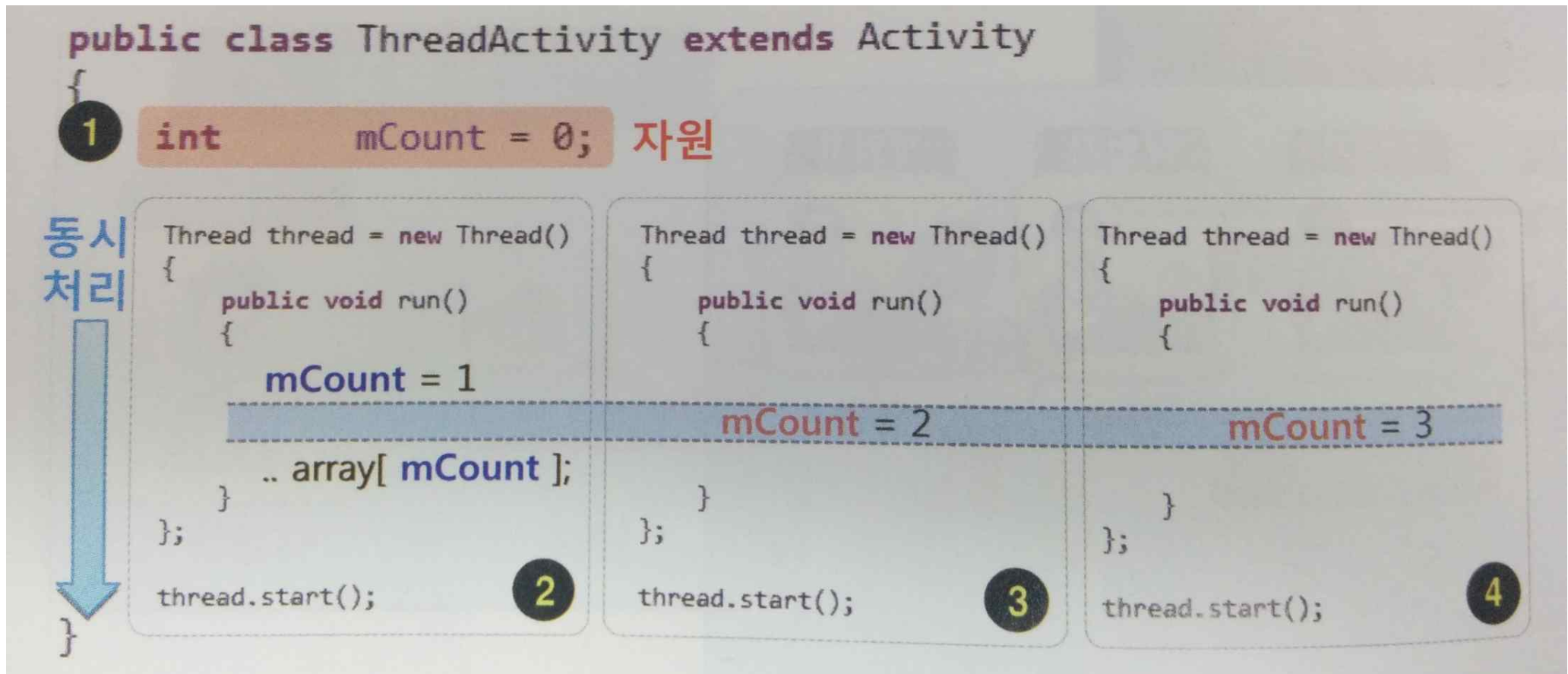
- 멀티 프로세스, 멀티 스레드 환경
- 안드로이드는 리눅스 커널을 이용하고 있기 때문에 안드로이드 앱은 리눅스 운영체제의 프로세스를 기반으로 실행된다
- 기본적으로 하나의 앱을 실행하면 하나의 프로세스로 실행된다
 - 애플리케이션 패키지 이름 = 프로세스 이름
 - 애플리케이션의 서로 다른 component를 다른 프로세스에서 실행할 수 있다
- 자바의 멀티 스레드 기능을 그대로 지원한다



자바 스레드

```
Thread thread = new Thread()  
{  
    public void run()  
    {  
         // 처리해야 할 일  
    }  
};  
  
thread.start();
```


프로세스의 자원을 공유하는 멀티 스레드



자바 synchronized 키워드

메인 스레드와 작업 스레드

메인 스레드

- 안드로이드에서 앱이 실행될 때 기본으로 생성되는 스레드
 - 각종 생명주기 함수들을 처리, 화면에 그림을 그리는 역할
 - 별도의 스레드를 생성하지 않는 한 동일한 프로세스에서 실행되는 모든 component는 메인 스레드에서 실행됨

Ch13MainThreadTest - [D:\GitRepo\MobileProgramming_2016-2\11_Process_Thread\Ch13MainThreadTest] - [app] - ...#app#src#main#java#kr#wac#koreatech#swkang#ch13mainthreadtest#MainActivity.java - Android Stu...

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

Ch13MainThreadTest app src main java

Android

app

- manifests
- java
 - kr.ac.koreatech.swkang.ch13mainthreadtest
 - MainActivity
- res
- Gradle Scripts

Android Device Monitor

File Edit Run Window Help

Quick Access

DDMS

Devices

| Name | Online | ? | [6.0, deb... |
|--------------------------------------|--------|--------------|--------------|
| emulator-5554 | Online | | |
| com.google.android.googlequicksearch | 1827 | 8600 | |
| com.svox.pico | 16163 | 8601 | |
| com.google.android.googlequicksearch | 1893 | 8602 | |
| com.android.systemui | 1638 | 8603 | |
| com.google.android.gms | 2246 | 8604 | |
| system_process | 1543 | 8605 | |
| com.android.defcontainer | 16135 | 8606 | |
| kr.ac.koreatech.swkang.ch13mainthrea | 8680 | 8607 / 87... | |
| com.google.android.googlequicksearch | 2090 | 8608 | |
| com.android.inputmethod.latin | 1835 | 8609 | |
| com.android.deskclock | 2380 | 8610 | |
| com.android.phone | 1877 | 8611 | |
| android.process.acore | 1947 | 8612 | |
| com.google.android.gms.persistent | 1852 | 8613 | |
| com.google.process.gapps | 2014 | 8614 | |
| com.google.android.gms.unstable | 2686 | 8615 | |

Threads

| ID | Tid | Status | utime | stime | Name |
|----|------|----------|-------|-------|-------------------------|
| 1 | 8680 | Runnable | 2 | 13 | main |
| *2 | 8685 | Wait | 0 | 0 | Signal Catcher |
| *3 | 8686 | Runnable | 0 | 1 | JDWP |
| *4 | 8690 | Monitor | 0 | 1 | HeapTaskDaemon |
| *5 | 8689 | Wait | 0 | 0 | FinalizerWatchdogDaemon |
| *6 | 8688 | Wait | 0 | 0 | FinalizerDaemon |
| *7 | 8687 | Wait | 0 | 0 | ReferenceQueueDaemon |
| 8 | 8691 | Runnable | 0 | 0 | Binder_1 |

Refresh Tue Nov 15 13:17:38 KST 2016

```
at android.os.MessageQueue.nativePollOnce(Native Method)
at android.os.MessageQueue.next(MessageQueue.java:323)
at android.os.Looper.loop(Looper.java:135)
at android.app.ActivityThread.main(ActivityThread.java:5417)
at java.lang.reflect.Method.invoke(Native Method)
at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:726)
at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:616)
```

Run app

```
I/MobileProgramming: Current count: 2
I/MobileProgramming: Current count: 3
I/MobileProgramming: Current count: 4
D/OpenGLRenderer: Use EGL_SWAP_BEHAVIOR_PRESERVE
[ 11-15 04:17:16.170 8680: HostConnection::get() New H
[ 11-15 04:17:16.230 8680: HostConnection::get() New H
I/OpenGLRenderer: Initialized EGL, version 1.4
```

LogCat

Console

Saved Filters + -

All messages (no filters)

| L... | Time | PID | TID | Application | Tag | Text |
|------|--------------------|-------|-------|---------------------|-----|------------------------------|
| I | 11-15 04:17:18.733 | 1852 | 1867 | com.google.andro... | art | Debugger is no longer active |
| I | 11-15 04:17:18.733 | 16135 | 16141 | com.android.defc... | art | Debugger is no longer active |
| I | 11-15 04:17:18.734 | 8680 | 8686 | kr.ac.koreatech... | art | Debugger is no longer active |
| I | 11-15 04:17:18.734 | 2686 | 2692 | com.google.andro... | art | Debugger is no longer active |

88M of 594M

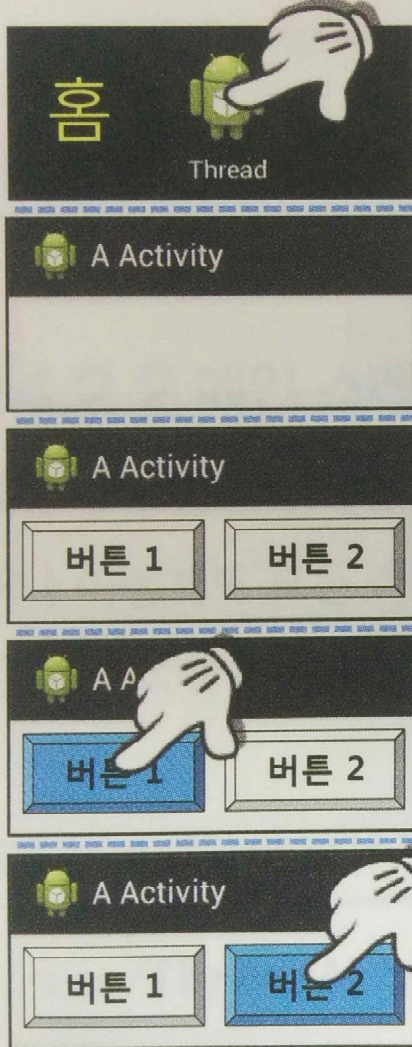
Instant Run applied code changes and restarted the app. // (Dont show again) (a minute ago)

27:1 CRLF+ UTF-8+ Context: <no context>

메인 스레드의 한계

- 모든 작업을 메인 스레드에서 할 수 없음
- 긴 시간이 걸리는 작업을 메인 스레드에서 하는 경우 어떤 일이 발생하게 되나?
- 예제
 - 예제 프로젝트 이름: 10_Process_Thread\Ch13MainThreadTest
 - onCreate() 함수에서 4초 동안 1초에 1씩 카운트하여 TextView에 출력하는 작업
 - 예상 결과는?

● 사용자와 상호 작용



메인 스레드

액티비티 생성 및 실행

생명주기 함수 처리

- onCreate(),
- onStart(),
- onResume()

화면에 뷰를 그린다.

사용자의 터치 이벤트에 반응한다.
버튼 1 뷰의 onClick()

사용자의 터치 이벤트에 반응한다.
버튼 2 뷰의 onClick()

3



긴 작업 수행

빨리 뷰들을 화면에
그려야하는데... T-T



6



긴 작업 수행

빨리 다음 사용자
터치 이벤트를
처리해야 하는데...

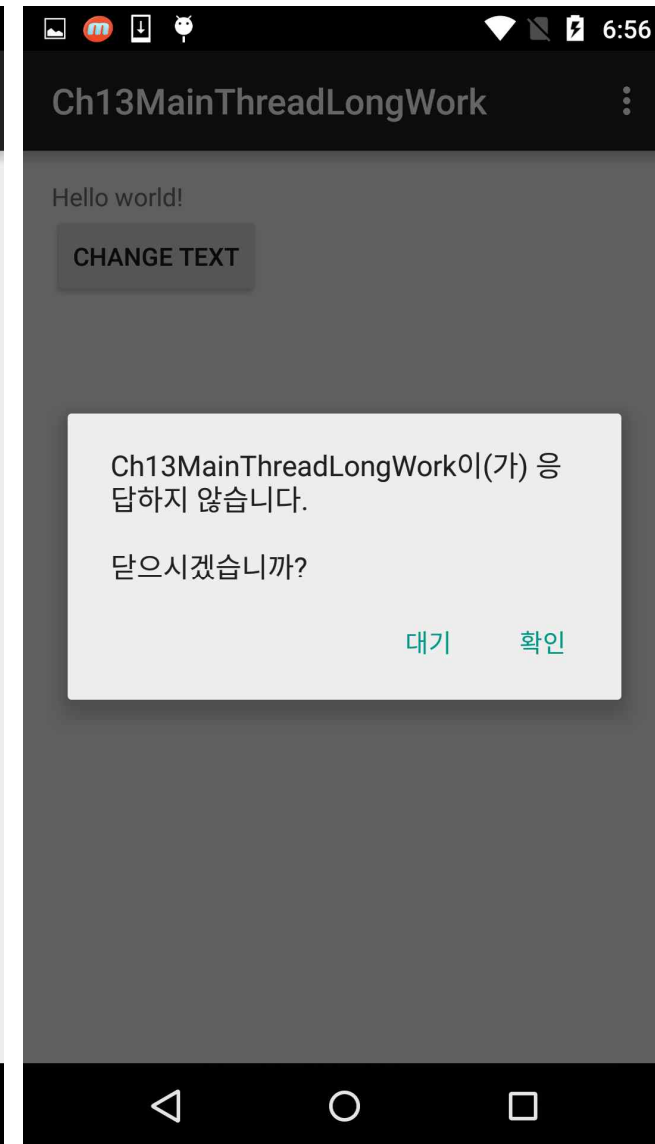
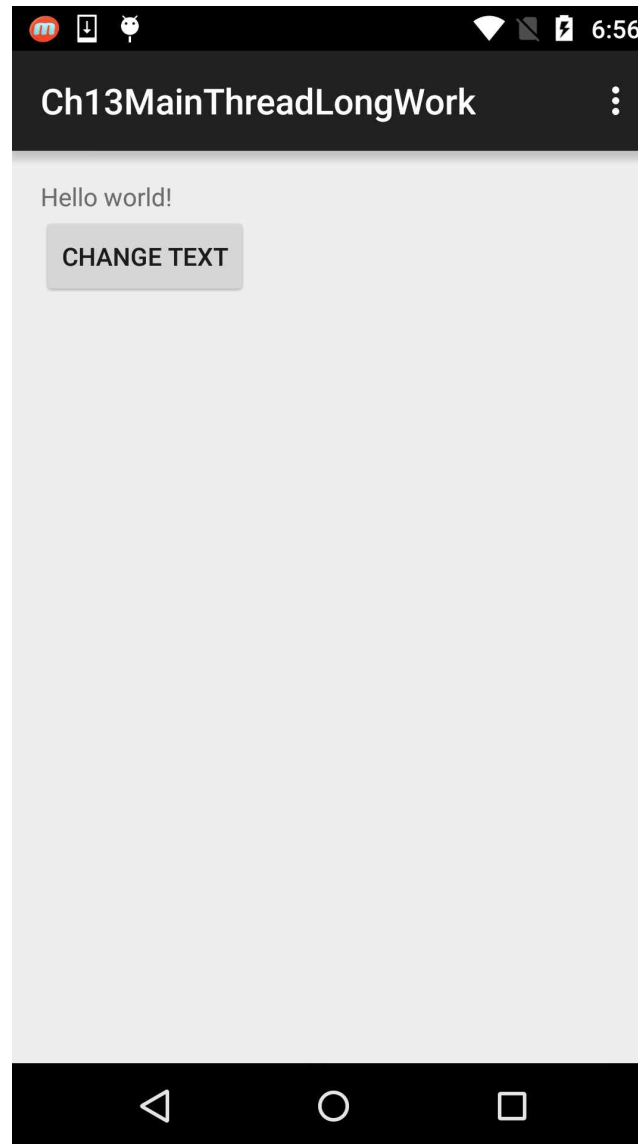


메인 스레드와 ANR

- 메인 스레드의 중요 임무
 - 사용자와 상호 작용하는 작업을 처리 → “UI 스레드”라고도 불림
 - 사용자의 UI 이벤트에 즉각적으로 반응해야 함
 - 그 결과를 화면에 끊김 없이 그려야 함
- 메인 스레드에서는 긴 시간이 걸리는 작업을 처리하면 안 된다
 - 메인 스레드가 긴 작업으로 잠기게 되는 경우
 - 사용자가 앱 사용 중에 화면이 뚝뚝 끊김
 - 각종 터치, 키 이벤트에 반응하지 않음
- ANR (Application Not Responding) 팝업
 - 메인 스레드가 잠기면 강제로 앱을 종료하기 위한 팝업 표시
 - 5초 이내에 입력 이벤트에 대한 응답이 없을 때
 - BroadcastReceiver가 10초 이내에 작업을 마치지 않을 때

ANR 예제

- 예제 프로젝트 이름
 - 10_Process_Thread\
Ch13MainThreadLongWork
- 버튼의 onClick 콜백 함수
 - 긴 시간 작업
 - 1초간 sleep 하며
20초 동안 카운트



메인 스레드에서 네트워크 사용

- 안드로이드는 앱이 끊김 없이 원활히 동작하도록 메인 스레드에서 처리할 수 없는 일 정의
 - 디스크 IO (디스크 읽기/쓰기)
 - 네트워크 사용
- 특히 네트워크 사용은 네트워크 혹은 서버의 상태에 따라 소모되는 시간을 유추하기가 힘들고 시간이 많이 소모될 확률이 높다
 - 네트워크 사용을 강제로 막고 있음
- Exception 발생
W/System.err : android.os.NetworkOnMainThreadException

메인 스레드에서 네트워크 사용

Activity의 onCreate() 메소드에 아래와 같은 코드를 넣고 실행해보자.

```
URL url;
try {
    url = new URL("http://www.google.com");
    HttpURLConnection urlConnection = (HttpURLConnection) url.openConnection();
    InputStream in = new BufferedInputStream(urlConnection.getInputStream());

} catch (Exception e) {
    e.printStackTrace();
}
```

작업 스레드

- 메인 스레드(UI 스레드)를 block 시키지 않기 위해서, 긴 시간이 걸리는 작업을 할 필요가 있는 경우 별도의 스레드를 생성하여 수행한다
- 자바의 스레드 모델 이용
 - Thread 클래스를 상속받는 클래스로 구현하는 방법
 - Runnable 인터페이스를 구현하는 방법

작업 스레드 이용 예제 1

- Thread 클래스를 상속받는 클래스로 구현하는 방법
 - 예제 프로젝트 이름: 10_Process_Thread\Ch13ThreadClass
 - 1초간 sleep 하며 20초 동안 카운트
 - 버튼을 누르면 현재 카운트 값을 TextView에 표시
- 앞의 예제의 경우 버튼을 눌렀을 때 반응을 하지 않았지만
이 예제에서는 버튼을 눌렀을 때 바로 카운트 값을 화면에 표시함

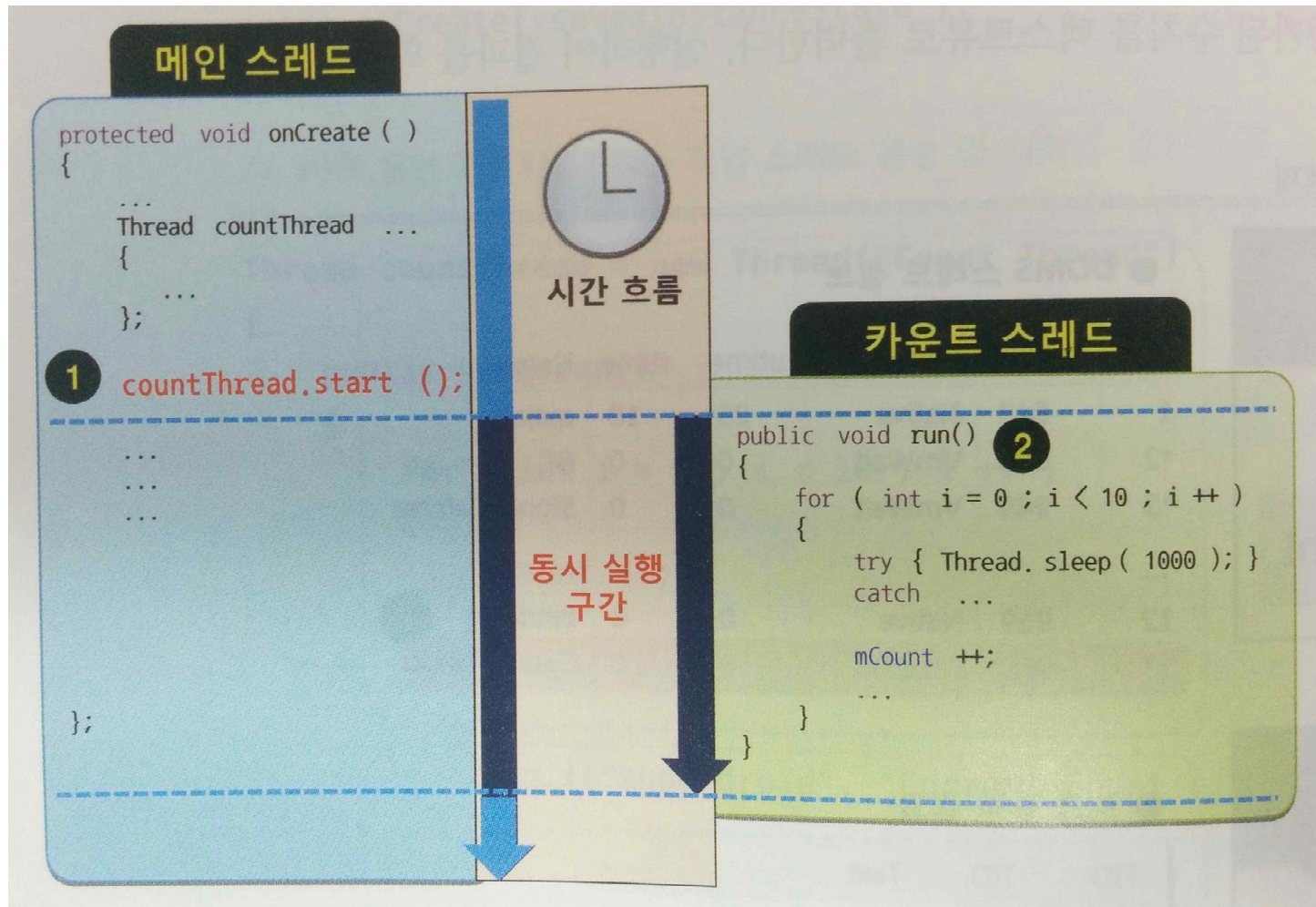
주요 code snippet

```
class CountThread extends Thread {  
    public void run() {  
        int i = 0;  
        for(i=0; i < 20 && running; i++) {  
            mCount++;  
  
            try {  
                Thread.sleep(1000);  
            } catch (InterruptedException e) {  
                e.printStackTrace();  
            }  
            Log.d("Mobile Programming", "Thread: time=" + mCount);  
        }  
    }  
}
```

```
@Override  
protected void onStart() {  
    super.onStart();  
    w = new CountThread();  
    running = true;  
    w.start();  
}
```

```
public void onClick(View viewW) {  
    tv.setText("Count: " + mCount);  
}
```

메인 스레드와 작업 스레드의 동시 실행



작업 스레드 예제 2

- Runnable 인터페이스를 구현하는 방법
 - Thread 클래스의 run() 함수와 동일하게 Runnable 인터페이스의 run() 함수를 구현하면 됨

- Thread 객체 생성 및 시작

```
Thread w = new Thread(new Runnable() {  
    public void run() {  
        ...  
    }  
});  
w.start();
```

작업 스레드 주의점

- 안드로이드에서 뷰의 내용을 변경하여 화면을 갱신하는 작업은 메인 스레드(UI 스레드)만 할 수 있다
- 작업 스레드에서는 화면에 표시되는 뷰 (프로그레스바, 텍스트 뷰, 이미지 뷰 등)를 직접 변경하면 안 된다
 - UI 툴킷은 thread-safe 모드가 아님
 - 즉, 멀티 스레딩 환경에서 UI 조작 시 동기화 문제가 발생할 수 있음
- Thread safe
 - 멀티 스레드 프로그래밍에서 일반적으로 어떤 함수나 변수, 객체가 여러 스레드에 의해서 동시에 접근이 이루어지더라도 프로그램 실행에 문제가 없음
 - 하나의 함수가 한 스레드로 호출되어 실행 중일 때, 다른 스레드가 그 함수를 호출하여 동시에 실행되어도 각 스레드에서 함수 실행 결과가 올바르게 나온다

작업 스레드에서 직접 UI 조작

- 예제

- 앞의 Thread 클래스 예제에서

`mCount++`; 다음에 아래와 같이 TextView의 텍스트를 변경하는 코드를 추가한 후 실행해보자

```
tv.setText("Count: " + mCount);
```

작업 스레드에서 직접 UI 조작

- Exception 발생

- 실행 결과

11-15 04:11:28.207 6144-

6158/kr.ac.koreatech.swkang.ch13threadclass E/AndroidRuntime:
FATAL EXCEPTION: Thread-185

Process: kr.ac.koreatech.swkang.ch13threadclass, PID: 6144

android.view.ViewRootImpl\$CalledFromWrongThreadException:
Only the original thread that created a view hierarchy can touch its
views.