

**git clone -b 05-async-1**

**<https://github.com/attrecto/Attrecto-Academy-Android.git>**

Ne manuálisan töltsd le a repót, mert később nem fogsz tudni  
könnyen branchet váltani!



# Párhuzamosítás

Demkó Gábor  
gabor.demko@attecto.com

# MIRŐL LESZ SZÓ?

---

## Első alkalom

- Párhuzamosítási feladatok és lehetőségek elméleti áttekintése
- Alkalmazás/process életciklus áttekintése (NEM azonos az Activity életciklussal)
- Példakód kipróbálása alapvető párhuzamosítási feladatokra

## Második alkalom

- Komplexebb gyakorlati példa az alábbi elemekkel:
  - hálózati kommunikáció
  - parsolás
  - párhuzamosítás
  - eredmény kijelzése
  - képkezelés
  - hibakezelés
  - tipikus hibák
  - háttérfolyamat leállítása



# MIÉRT KELL PÁRHUZAMOSÍTANI?

---

Alapértelmezés szerint a kódunk a Main threaden, másnéven UI threaden fut

Ha egy művelet sor sokáig tart, akkor az alkalmazás felhasználói felülete a futás alatt nem reagál, lefagy (ANR üzenet, Application Not Responding)

A hosszan tartó műveleteket párhuzamosítani kell, azaz egy új háttérszálon kell végrehajtani a műveletet

Specialitás: A felhasználó felülethez csak az UI thread nyúlhat!



# ALKALMAZÁS/PROCESS ÉLETCIKLUS

---

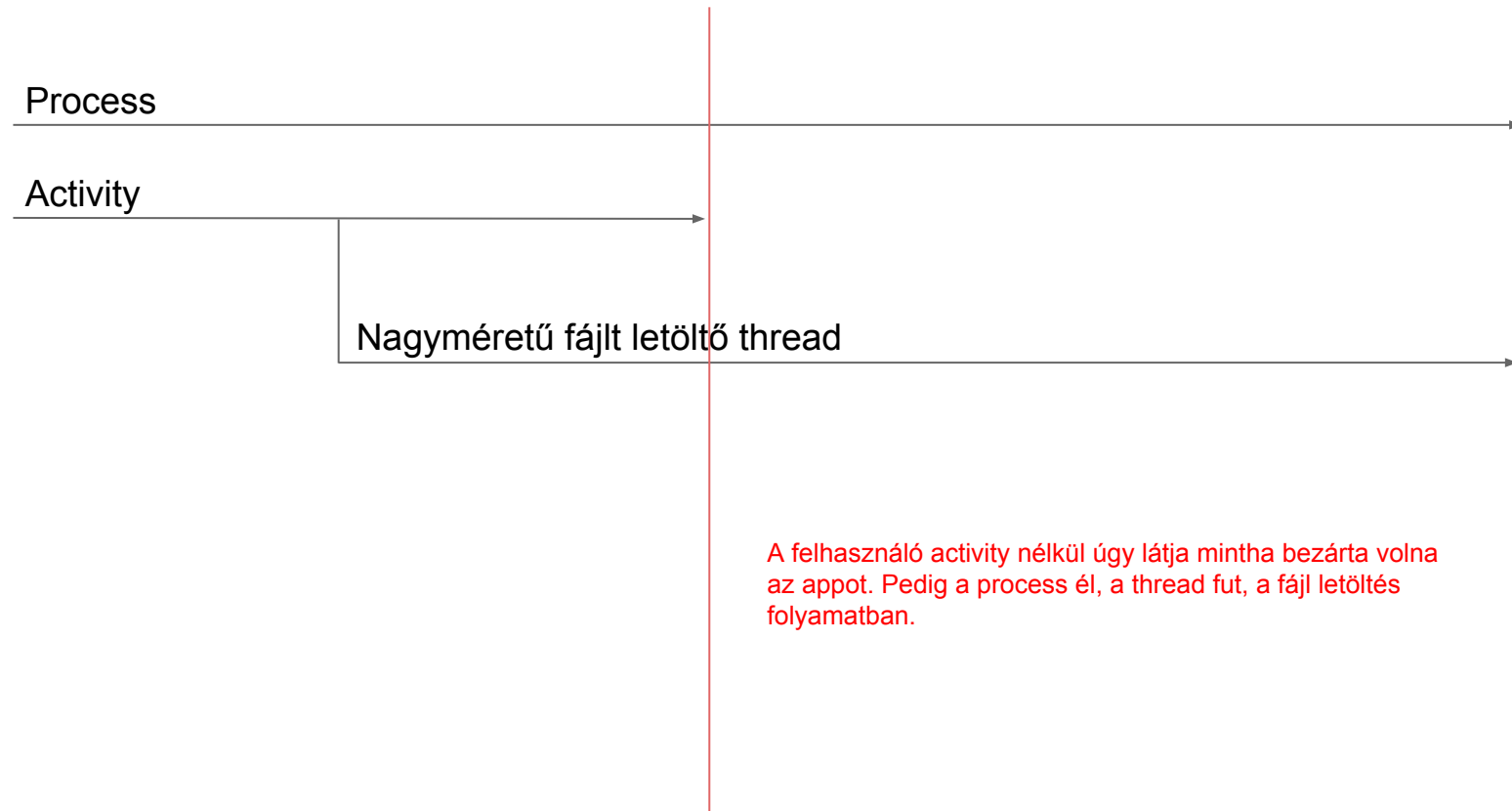
- egy alkalmazás alapvetően egy process (ettől csak extrém esetben kell eltérni)
- a process életciklusát, közvetlenül nem tudjuk befolyásolni, az operációs rendszer indítja el és állítja le
- a process életciklusa elsősorban a benne futó komponensektől függ (Activity, Service, BroadcastReceiver)

Ha egy processben megszűnik minden Activity, Service és BroadcastReceiver, akkor az Android rendszer bármikor megszüntetheti a processt. Akkor is ha az még munkát végez!

A háttérzálak alapvetően nem kötődnek Activity, Service vagy BroadcastReceiver-hez. Nem szűnnek meg ha pl. bezárjuk az Activity-t honnan indítottuk. De ha a rendszer megszünteti a process természetesen a zálak is megszűnnek.



# Példa



A rendszer kilövi az activity nélküli  
processt, a fájl letöltés megszakad

A felhasználó activity nélkül úgy látja mintha bezárta volna  
az appot. Pedig a process él, a thread fut, a fájl letöltés  
folyamatban.



# Párhuzamosítási lehetőségek

---

Több lehetőség is van, ezek közül mi két módszert vizsgálunk meg, ami a legtöbb helyzetben könnyen alkalmazható.

- AsyncTask: rövidebb feladatokra, egyszerű szerverhívásokra, miközben az alkalmazás képernyői láthatóak
- IntentService: hosszabb feladatokra, főleg ha közben a felhasználó bezárhatja az alkalmazást (az alkalmazás képernyőit)



# AsyncTask

```
new AsyncTask<Params, Progress, Result>(){  
  
    @Override  
    protected void onPreExecute() {  
        //UI szálon fut, még a háttérszál előtt  
    }  
  
    @Override  
    protected Result doInBackground(Params... params) {  
        //háttérszálon fut, itt végezzük el a párhuzamosítandó feladatot  
        return new Result();  
    }  
  
    @Override  
    protected void onPostExecute(Result result) {  
        //UI szálon fut, itt már megkapjuk a háttérszál eredményét  
    }  
  
    @Override  
    protected void onCancelled(Result result) {  
        //megszahítás esetén fut a UI szálon  
    }  
};
```





# IntentService

```
public class DownloadService extends IntentService {  
  
    public DownloadService() { super(name: "DownloadService"); }  
  
    @Override  
    protected void onHandleIntent(@Nullable Intent intent) {  
        //háttérszálon fut ez a metódus  
  
        //a sima Service-el ellentétben az IntentService ezen fellül:  
        //- magától leáll  
        //- sorbaállítja a kéréseket  
  
        //ha a lehető legerősebben védeni akarjuk a Service-t a leállítástól:  
        startForeground(id: 1, new Notification());  
    }  
}
```



# Köszönöm a figyelmet!



**Attrecto Zrt.**  
**Attrecto Next Tech Digital Solutions**

H-9024 Győr, Wesselényi str. 6.  
info@attrecto.com

