* Android emulator nedir? nasıl çalışır? nasıl kullanılır?
  + Android emulator, fiziksel bir aygıta gerek kalmadan Android API, sensörler veya uygulamaları denememize fırsat veren bir programdır. Her çalışan emulatorun bir tane Android Sanal Aygıt (Android Virtual Device - avd) kurulumuna ihtiyacı vardır. AVD; android versiyonu, taklit edilen cihazın fiziksel özellikleri ve sabit sürücüsü gibi bilgileri içerir.
  + AVD oluşturmak için aygıt tanımı dosyası ve sistem imaj dosyası gerekir. (SDK Manager üzerinden) Bunlar temin edildikten sonra bir AVD oluşturulabilir. Bunu hem Android Studio üzerinden, hem de komut satırından Avdmanager çalıştırılarak yapabiliriz.
  + (avdmanager create avd -n ISIM -k SIS\_IMAJ -d AYGIT)
  + Oluşturulan AVD’yi yine Android Studio yada komut satırından emulator ile açabiliriz.
  + (emulator -avd ISIM)
* ADB android emulator ile birlikte çalışabilir mi?
  + Evet, ADB emulatorler ile çalışabilir. ADB’nin varsayılan ayarları, 5554-5586 portları arasında 16 tane emulator ile çalışabilir. Emulatorler, varsayılan olarak zaten bu portları kullanacaktır. ADB ayarlarından $ADB\_LOCAL\_TRANSPORT\_MAX\_PORT’ın değeri, emulator sayısını belirlemek için değiştirilebilir.
  + “Adb devices -l” ile adb sunucusunun farkında olduğu emulatorler görülebilir.
* ADB shell ile UE uçak modu enable/disable nasıl yapılır?
  + Eskiden “settings put global airplane\_mode\_on 1(ya da 0); am broadcast -a android.intent.action.AIRPLANE\_MODE” komutu ile UE uçak modu enable/disable edilebiliyormuş, ama bu Android 7’den itibaren kaldırılmış. Artık bu komutu çalıştırabilmek için root yetkisi gerekiyor.
  + adb shell "su 0 settings put global airplane\_mode\_on 1 && su 0 am broadcast -a android.intent.action.AIRPLANE\_MODE"
  + Alternatif olarak shell üzerinden key eventler çalıştırarak, UI üzerinden uçak modu ayarları açılıp, menü seçeneklerinde teker teker aşağı inilip uçak modu açılabilir, ama UI düzeni devamlı değişebileceği için bu çok güvenli bir yöntem değil.
* ADB shell ile 4G/5G şebeke kayıtlandı bilgisi nasıl görülür?
  + “dumpsys” aracı, UE ile ilgili bir çok bilgiye erişmemizi sağlar. UE’nin o anki şebeke bilgilerine erişmek için shell’de bu aracı kullanabiliriz.
  + dumpsys telephony.registry
  + Burada önümüze bir çok bilgi düşüyor. “mServiceState” diye başlayan satırda “getRilVoiceRadioTechnology=” kısmı bize EDGE(2G), HSPA(3G), LTE(4G) veya NR-SA(5G) bilgisini veriyor. Sadece bu bilgiye erişmek için üstteki komutu grep kullanarak sadeleştirebiliriz.
  + dumpsys telephony.registry | grep 'mServiceState' (daha da sadeleştirilebilir)
  + Ayrıca, “dumpsys telephony.registry” altta küçük bir log veriyor ve burada EDGEden LTEye gibi geçişleri görmek mümkün.
* ADB shell ile iperf komutu koşturabilir miyim?
  + “iperf”, Android’de varsayılan bir program olarak gelmez. Github’da “iperf”in source dosyalarından binary oluşturup, adb üzerinden bu binaryi telefona atıp, shell’den bunu çalıştırmayı tavsiye etmişler. Bu yöntem ile adb shell üzerinden iperf çalıştırmak mümkün.
  + <https://github.com/KnightWhoSayNi/android-iperf>
  + Buradan hazır binary dosyasını indirip, “adb push LOCALPATH/iperf3.10.1 /data/local/tmp/iperf3.10.1” ile dosyayı telefona atıp, bu dosyanın içinden “adb shell ./iperf ARGUMANLAR” komutuyla bu program çalıştırılabilir.