Lab 8 – ADB

Topic: Using ADB Shell
See Sakai for deadline and submission instructions.

Points: 10

#	Task	Check
1	Connect your phone to your laptop over USB and test if the ADB is working by issuing this command from a command line tool: • **`adb devices -l`**	1
2	From the command line interface, issue the following command to get information on the network interfaces on your phone: • **`adb shell ifconfig`** • Try the other way of running an ADB shell command, i.e. issue **`adb shell`** to get inside the phone's shell, and then enter **`ifconfig`**.	1

3	Notice the output carefully. If WiFi is enabled, you will see an interface called the **`wlan0`**. Notice its IP address, TX, and RX packets.
	
	`wlan0 Link encap:UNSPEC`
	`inet addr:192.168.0.8 Bcast:192.168.0.255 Mask:255.255.255.0`
	`inet6 addr: 2606:a000:aec0:4b00:f901:24ec:bf75:580a/64 Scope: Global`
	`inet6 addr: 2606:a000:aec0:4b00:faa9:d0ff:fe50:53fd/64 Scope: Global`
	`inet6 addr: fe80::faa9:d0ff:fe50:53fd/64 Scope: Link`
	`UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1`
	`RX packets:2751738 errors:0 dropped:0 overruns:0 frame:0`
	`TX packets:2151205 errors:0 dropped:0 overruns:0 carrier:0`
	`collisions:0 txqueuelen:1000`
	`RX bytes:2666839418 TX bytes:735159720`
		1

4	Using Android studio, create an App that executes the **`ifconfig`** command using the **`Runtime`** and **`Process`** classes. Note that you have to give your app the **`android.permission.INTERNET`** permission in the AndroidManifest.xml.	2
5	Modify you app so that it parses the output of **`ifconfig`** and shows three information (IP address, TX, and RX packets) in three **`TextView`**.	3
6	Modify the app further so that we get real-time updates on TX and RX packets. You may have to use a thread to periodically (e.g. every second or two) issue **`ifconfig`** command, parse it, and update the text views.	2