Unix Project

Introduction

For our project, we've decided to package a python script to debian. At first we were going to make an operating system catered to gamers, however we did not have much time left in the semester to do so and the second idea was to package a game to debian but got blocked on the dependencies. As for the python packaging, here are our steps into completing the process:

This is to create the directory to hold the python script

```
vboxuser@vboxuser:~$ ls
Desktop  Documents  Downloads  Music  new_package  Pictures  Public  Templates  Videos
vboxuser@vboxuser:~$ mkdir unixproject_package
vboxuser@vboxuser:~$ cd  unixproject_package
vboxuser@vboxuser:~/unixproject_package$ mkdir macc
vboxuser@vboxuser:~/unixproject_package$ cd macc
vboxuser@vboxuser:~/unixproject_package/macc$ touch __init__.py
vboxuser@vboxuser:~/unixproject_package/macc$ touch macc.py
vboxuser@vboxuser:~/unixproject_package/macc$ ls
__init__.py  macc.py
vboxuser@vboxuser:~/unixproject_package/macc$
```

Here we change the mac address

```
root@vboxuser:~/unixproject_package/macc# sudo ifconfig enp0s3 down
root@vboxuser:~/unixproject_package/macc# sudo ifconfig enp0s3 hw ether 11:22:33:44:55:
66
SIOCSIFHWADDR: Cannot assign requested address
root@vboxuser:~/unixproject_package/macc# sudo ifconfig enp0s3 hw ether 00:66:22:77:11:
88
root@vboxuser:~/unixproject_package/macc# sudo ifconfig enp0s3 up
```

```
root@vboxuser:~/unixproject_package/macc# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        ether 00:66:22:77:11:88  txqueuelen 1000  (Ethernet)
        RX packets 37461  bytes 55550311 (52.9 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3038  bytes 237230 (231.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 61  bytes 7196 (7.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 61  bytes 7196 (7.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Creating a python script to automatically change the mac address

```
 GNU nano 7.2                          macc.py *
import subprocess

class Macchanger:
        def change_mac(self,interface,new_mac):
                subprocess.call(["sudo","ifconfig",interface,"down"])
                subprocess.call(["sudo","ifconfig",interface,"hw","ether", new_mac])
                subprccess.call(["sudo","ifconfig",interface,"up"])
                print("[+] Changing MAC adress of "+interface+" to "+new_mac)

^G Help      ^O Write Out  ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File  ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

The calling method written in python that executes the process

```
def get_arguments(self):
        parser = argparse.ArgumentParser()
        parser.add_argument("-i","--interface",dest="interface",help="Interfac>
        parser.add_argument("-m","--mac",dest="new_mac",help="New MAC address")
        options = parser.parse_args()
        if not options.interface:
                parser.error("[-] Please specify an interface")
        elif not options.new_mac:
                parser.error("[-] Please specify a new mac address")
        return options
```

Executing the method through terminal

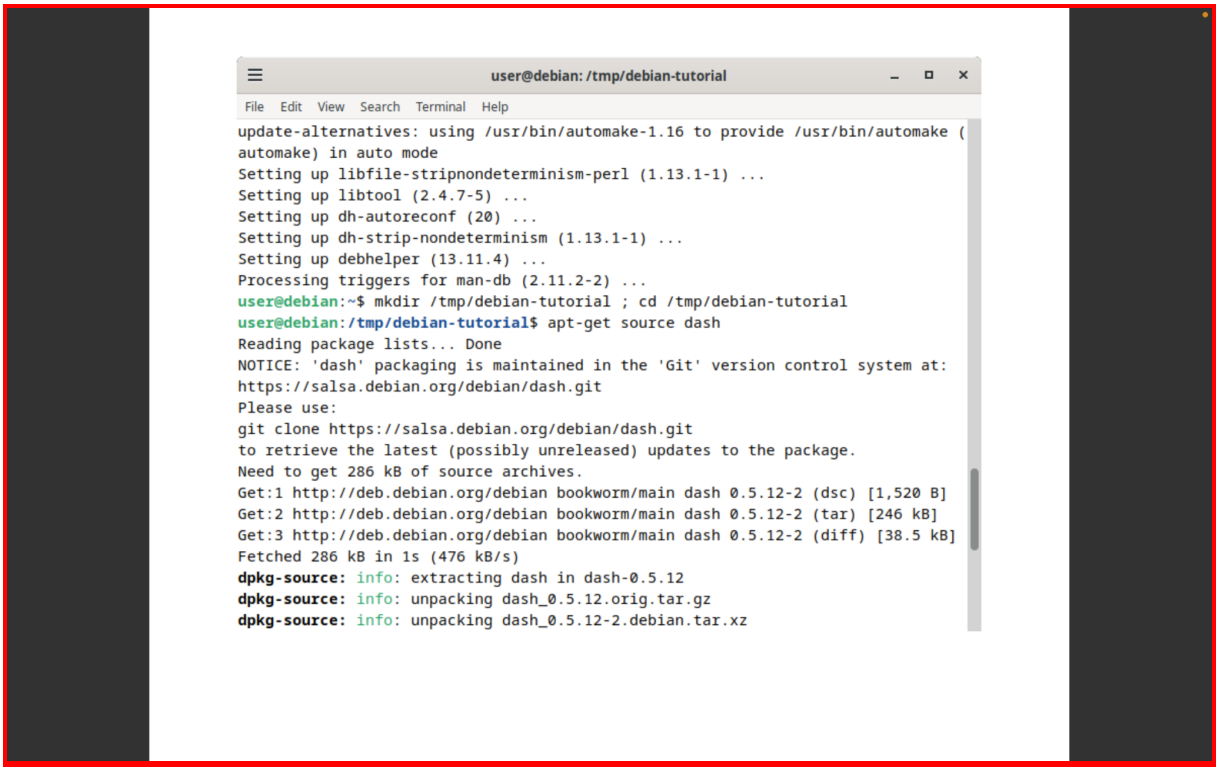
```
root@vboxuser:~/unixproject_package/macc# python3 macc.py -i enp0s3 -m 00:55:22:44:55:6
6
Traceback (most recent call last):
  File "/root/unixproject_package/macc/macc.py", line 29, in <module>
    main()
  File "/root/unixproject_package/macc/macc.py", line 26, in main
    mac_changer.change_mac(options.interface,options.new_mac)
  File "/root/unixproject_package/macc/macc.py", line 19, in change_mac
    subprccess.call(["sudo","ifconfig",interface,"up"])
    ^^^^^^^^^^
NameError: name 'subprccess' is not defined. Did you mean: 'subprocess'?
root@vboxuser:~/unixproject_package/macc# nano macc.py
root@vboxuser:~/unixproject_package/macc# python3 macc.py -i enp0s3 -m 00:55:22:44:55:6
6
[+] Changing MAC adress of enp0s3 to 00:55:22:44:55:66
root@vboxuser:~/unixproject_package/macc#
```

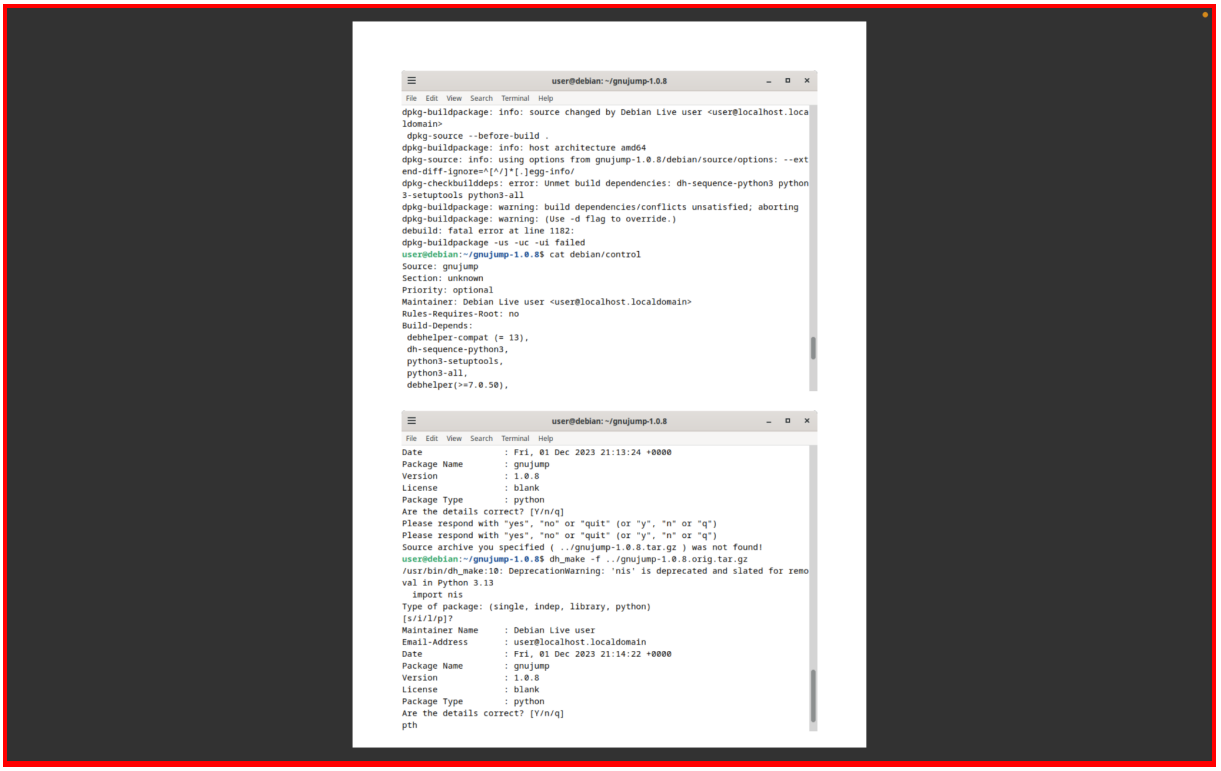
Licence and README to the project

```
root@vboxuser:~/unixproject_package/macc# touch setup.py; touch license; touch README.m
d; touch MANIFEST.in
root@vboxuser:~/unixproject_package/macc# ls
__init__.py  license  macc.py  MANIFEST.in  README.md  setup.py
```

Reasons why we decided to switch

When we decided to start to package a game we encountered issues with the build-dependencies portion as we weren't too sure what to adjust so that the dependencies would make the process continue. We were only able to complete the following parts:





What we learned in our project:

Learning how to package games and Python scripts for Debian involves gaining a comprehensive understanding of the Debian packaging process. In the course of our project, we learned how to package items to debian whether it be a game or a script. For games, this

process includes identifying and listing dependencies, specifying installation paths, and addressing any compatibility issues. Similarly, packaging Python scripts involves creating a Debian source package, defining dependencies, and adhering to Debian's policies for file organisation and system integration.

Main Struggles:

The main area we struggled with throughout the course of this project was creating the dependencies as we weren't too sure at the start with what needed to be put as a dependency. This was a major reason as to why we didn't go through with packaging a game due to the intensive dependencies it requires.

Demonstration