Documentation

The File-Directory Management System

Initial Attempt(Not Used in Final Code)

Initially the plan was to store the output of the ML-model in a temporary file and then call the python file with the content of the file as parameters.

```
os.system("/usr/bin/python3 final.py < tmp.txt")
```

Inside the python file the code

```
import os
import sys
cmd=""
code=""
def joiner():
    global cmd
    m=len(sys.argv)
    for i in range(1,m):
        cmd=cmd+sys.argv[i]+" "
    cmd=cmd.lower()
def replacer():
    global cmd
    t=""
    for i in cmd:
        if (i==',' or i==";"):
            t=t+" "
        else:
            t=t+i
    cmd=t
joiner()
replacer()
```

```
def executor(cmd):
    cmd=cmd.split()
    global code
    if (cmd[0]=="read"):
        code="./read "+cmd[2]
    elif(cmd[0]=="create" and cmd[1]=="file"):
        code="./create "+cmd[2]
    elif(cmd[0]=="create" and cmd[1]=="folder"):
        code="./makedir "+cmd[2]
    elif(cmd[0]=="copy" and cmd[1]=="file"):
        code="./copy "+cmd[2]+" "+cmd[3]
    elif(cmd[0]=="copy" and cmd[1]=="folder"):
        code="./makedir "+cmd[2]
    ...[Other cases]
    os.system(code)
executor(cmd)
```

ISSUE WITH THE APPROACH

Since os.system() will launch a new sh shell by default everytime so the shell is amnesiac in nature and cannot remember the final state and directory of the system.

Rectified Approach

```
def init():
    os.environ["pwd"]="/home/kali"
    os.system("export PATH=/usr/local/sbin:/usr/local/bin:/us
init()
```

init() function initialises the environment variable “pwd” and set the path to location of the compiled codes.

```
def replacer(cmd):
    t=""
    for i in cmd:
        if (i==',' or i==";"):
            t=t+" "
        else:
            t=t+i
    cmd=t
    return(cmd)
```

replacer() replaces punctuation caused due to pause in the audio.

```
def executor(cmd):
    t=""
    cmd=replacer(cmd)
    cmd=cmd.split()
    global code
    # print("pwd:",os.environ["pwd"])
    if(len(cmd)>2):
        cmd[2]=os.environ["pwd"]+"/"+cmd[2]
    if len(cmd)>3:
        cmd[3]=os.environ["pwd"]+"/"+cmd[3]
    if (cmd[0]=="read"):
        code="/home/kali/os/OS-Project/file_bin/read "+cmd[2]
    elif(cmd[0]=="create" and cmd[1]=="file"):
        t="Copied"
        code="/home/kali/os/OS-Project/file_bin/create "+cmd[
    elif(cmd[0]=="create" and cmd[1]=="directory"):
        t="Created"
        code="/home/kali/os/OS-Project/file_bin/makedir "+cmd
    elif(cmd[0]=="copy" and cmd[1]=="file"):
        t="Copied"
        code="/home/kali/os/OS-Project/file_bin/copy "+cmd[2]
    elif(cmd[0]=="copy" and cmd[1]=="directory"):
        t="Copied"
        code="/home/kali/os/OS-Project/file_bin/copy -r "+cmd
    elif(cmd[0]=="rename" ):
        t="Renamed"
        code="/home/kali/os/OS-Project/file_bin/move "+cmd[2]
    elif(cmd[0]=="remove" and cmd[1]=="file"):
        t="Removed"
        code="/home/kali/os/OS-Project/file_bin/remove "+cmd[
    elif(cmd[0]=="remove" and cmd[1]=="directory"):
        t="Removed"
        code="/home/kali/os/OS-Project/file_bin/remove -rf "+
    elif(cmd[0]=="change"):

        os.environ["pwd"]=cmd[2]

    elif(cmd[0]=="go"):
        os.environ["pwd"]=os.environ["pwd"]+"/.."
        print("backked")
    elif(cmd[0]=="list"):
        code="/home/kali/os/OS-Project/file_bin/list "+os.env
    else:
        print(cmd)
        return('Error')
    try:
        if(code!=""):
            print(code+" pwd:"+os.environ["pwd"])
            os.system(code+" >tmp.txt")
            code=""

            return(open("tmp.txt","r").read())

        code=""

    except:
        print(code)
        code=""
        return('Error')
```

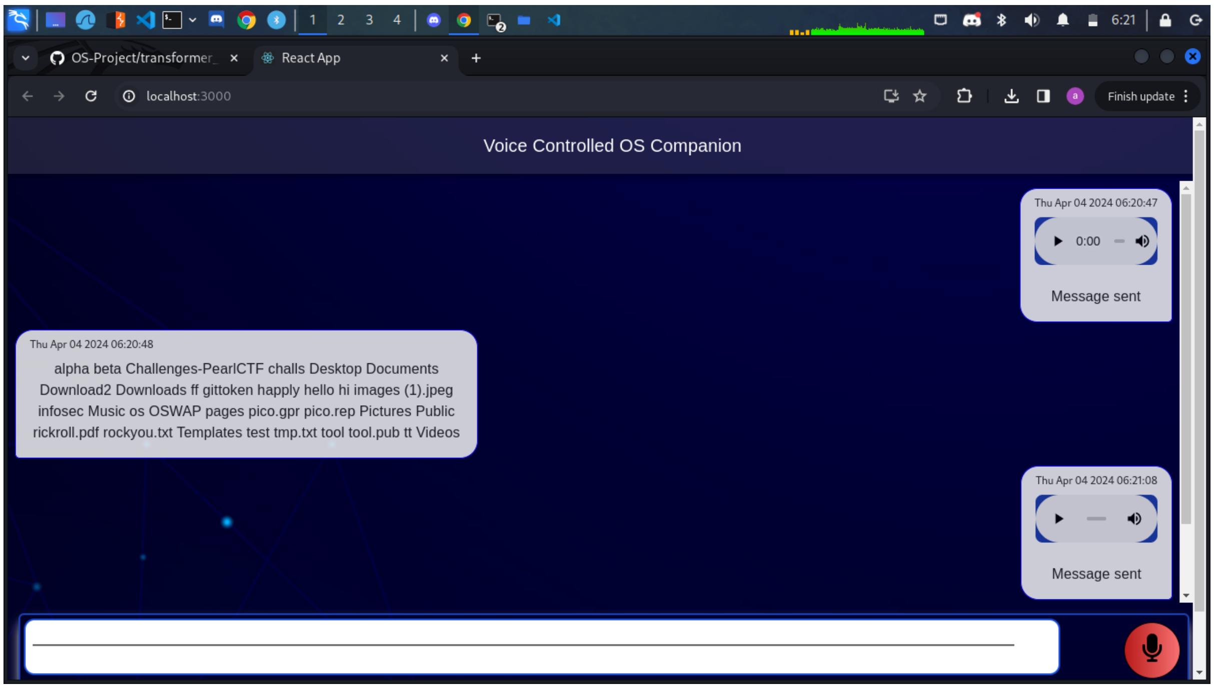
Finally executor() reads the environment variable “pwd” and converts all relative path to absolute path, executes the commands based on the case ans store the result to tmp.txt.

Instead of executing python file for every audio clip , I have integrated the file manipulation binary within the django and ML-Model function itself.

Advantages:

Removes the necessity of calling the python file every time , substantially reducing the workload.

As it stores the present working directory in an environment variable hence it ensures that the final state and directory remains consistent.



Thu Apr 04 2024 06:20:48

```
alpha beta Challenges-PearlCTF challs Desktop Documents Download2 Downloads ff gittoken happly hello hi images (1).jpeg infosec Music os OSWAP pages pico.gpr pico.rep Pictures Public rickroll.pdf rockyou.txt Templates test tmp.txt tool tool.pub tt Videos
```