import os

import shutil

from datetime import datetime

import subprocess

import time

#start the rclone mount to OneDrive from the connectrclone.sh file

#Global Commands

# Get the day of the week and time of day

dow = datetime.now().strftime('%A')

tod = datetime.now().strftime("%-I%p") #format 7PM

#get the min from the time

mod = datetime.now().strftime('%M')

# Set the flag to check if VLC process is already open

print("Day of the Week: "+dow+", time of day: "+tod+", min of day: "+mod)

vlc\_process = None

#Set the current file name

current\_file\_mtime = ""

current\_file\_name = ""

def get\_recent\_file():

    day\_of\_week = datetime.now().strftime('%A')

    time\_of\_day = datetime.now().strftime("%-I%p")

    min\_of\_day = int(datetime.now().strftime('%M'))

    session\_times = ["7AM", "8AM", "9AM"] #session announcement for morning session locations 7-10am

    opening\_closing\_session\_times = ["5PM","6PM"] #saturday/thursday opening/closing session location 5-7pm

    evening\_sessions = ["Saturday","Thursday"] #evening session days

    celebration\_times = ["7PM","8PM","9PM"] #video playing times for thursday 7-10pm

    connected = False

    while connected is False:

        try:

            #if the time of day is around the time of session

            if time\_of\_day in session\_times:

                directory = "/home/fortadmin/OneDrive/" + day\_of\_week + "/Session"

            elif time\_of\_day in celebration\_times and day\_of\_week == "Thursday":

                directory = "/home/fortadmin/OneDrive/" + day\_of\_week + "/Celebration\_Video"

            elif time\_of\_day in opening\_closing\_session\_times and day\_of\_week in evening\_sessions:

                directory = "/home/fortadmin/OneDrive/" + day\_of\_week + "/Evening\_Session"

            else:

                # Construct directory path based on the day of the week

                directory = "/home/fortadmin/OneDrive/" + day\_of\_week + "/General\_Announcements"

                if not os.listdir(directory):

                    return "/home/fortadmin/Desktop/Pi-Kiosk-App/WelcometoCamp.mp4"

        except OSError as e:

            #if the file is not found - wait for connection to happen

            print("File not found: check internet connection")

            time.sleep(60)

            continue #run the last downloaded file

        finally:

            if not os.path.isdir(directory) or not os.listdir(directory):

                directory = "/home/fortadmin/OneDrive/" + day\_of\_week + "/General\_Announcements"

                if not os.listdir(directory):

                    return "/home/fortadmin/Desktop/Pi-Kiosk-App/WelcometoCamp.mp4"

            # Get the list of .mp4 files in the directory

            files = sorted((os.path.join(directory, f) for f in os.listdir(directory) if f.endswith('.mp4')), key=os.path.getmtime, reverse=True)

            connected = True

    # Return the path of the most recent .mp4 file, or None if no such file exists

    return files[0] if files else None

def stop\_vlc():

    global vlc\_process

    if vlc\_process and vlc\_process.poll() is None:

        # Terminate the VLC process gracefully

        vlc\_process.terminate()

        vlc\_process.wait()

def start\_vlc(recent\_file):

    global vlc\_process

    # Start VLC with the recent file

    try:

        vlc\_process = subprocess.Popen(['vlc', recent\_file, '--loop', '--fullscreen', '--no-video-title-show'])

    except:

        print("There was a error opening the file")#play the welcome to camp file while the file is downloading

        if vlc\_process is None or vlc\_process.poll() is not None:

            #display the for logo if there are no files in the days directory

            start\_vlc("/home/fortadmin/Desktop/Pi-Kiosk-App/WelcometoCamp.mp4")

        #if the welcome to camp file is already running then dont change anything

        elif recent\_file == "/home/fortadmin/Desktop/Pi-Kiosk-App/WelcometoCamp.mp4":

            print("The file is the same")

        else:

            #else start the welcome to camp

            time.sleep(60)

            stop\_vlc()

            time.sleep(4)

            start\_vlc("/home/fortadmin/Desktop/Pi-Kiosk-App/WelcometoCamp.mp4")

def savefile\_desktop(recent\_file):

    print("Recent file: " + recent\_file)

    #copy the most recent file to the desktop

    filename = "/home/fortadmin/Desktop/Pi-Kiosk-App/running\_file.mp4"

    try:

        #copy the new file to the desktop file

        dest = shutil.copyfile(recent\_file, filename)

        #pause the system to let it catch up

        time.sleep(5)

        print("File saved: ", dest)

    except shutil.SameFileError:

        print("Source and destination file are the same.")

    except:

        print("There was another error")

    return filename

while True:

    try:

        #get the most recent created file and get the modification date

        recent\_file = get\_recent\_file() # day of the week info is found in the function

        print(type(recent\_file))

        recent\_mtime = os.path.getmtime(recent\_file)

        #check to see if the new file was made at a different time than the current running file

        if current\_file\_mtime != recent\_mtime or current\_file\_name != recent\_file:

            #if there is a running vlc process

            stop\_vlc()

            #save the new file to the desktop to play from

            filename = savefile\_desktop(recent\_file)

            #save the new files info to be compared later

            current\_file\_mtime = recent\_mtime

            current\_file\_name = recent\_file

        #if there is no vlc process running then start or restart the video

        if vlc\_process is None or vlc\_process.poll() is not None:

               # Start VLC with the recent file

               start\_vlc(filename)

        #if none of these if statements are triggered just continue on and do nothing

        # Sleep for a while before checking again (to avoid constant checking)

        time.sleep(60)

        print("After time")

    except OSError:

        print("There was an error in the file system")

        pass #continues to search for files

    except TypeError:

        print("There was no files in the directory")

        recent\_file = "/home/fortadmin/Desktop/Pi-Kiosk-App/WelcometoCamp.mp4"

         #if there is no vlc process running then start or restart the video

        if vlc\_process is None or vlc\_process.poll() is not None:

            #display the for logo if there are no files in the days directory

            start\_vlc(recent\_file)

        elif recent\_file == "/home/fortadmin/Desktop/Pi-Kiosk-App/WelcometoCamp.mp4":

            print("The file is the same")

        else:

            stop\_vlc()

            time.sleep(4)

            start\_vlc("/home/fortadmin/Desktop/Pi-Kiosk-App/WelcometoCamp.mp4")

        continue