raspberry-doc

huseyinerdal

October 2020

1 Hiding Cursor

I simply added a nocursor option as follows in the file (/etc/lightdm/lightdm.conf)

```
xserver-command = X -nocursor
```

2 Fast Boot Raspberry Pi 3

Editing the /boot/config.txt with the following changes:

```
# Disable the rainbow splash screen
disable_splash=1

# Disable bluetooth
dtoverlay=pi3-disable-bt

#Disable Wifi
dtoverlay=pi3-disable-wifi

# Overclock the SD Card from 50 to 100MHz
# This can only be done with at least a UHS Class 1 card
dtoverlay=sdtweak,overclock_50=100

# Set the bootloader delay to 0 seconds.
# The default is 1s if not specified.
boot_delay=0

# Overclock the raspberry pi. This voids its warranty.
# Make sure you have a good power supply.
force_turbo=1
```

Use systemd-analyze blame, systemd-analyze critical-chain to disable services I didn't need

```
sudo systemctl disable dhcpcd.service
sudo systemctl disable networking.service
sudo systemctl disable ssh.service
sudo systemctl disable ntp.service
sudo systemctl disable dphys-swapfile.service
sudo systemctl disable keyboard-setup.service
sudo systemctl disable apt-daily.service
sudo systemctl disable wifi-country.service
sudo systemctl disable hciuart.service
sudo systemctl disable raspi-config.service
sudo systemctl disable avahi-daemon.service
sudo systemctl disable triggerhappy.service
```

3 Raspberry Pi Connect Wifi

For this purpose, we can use python and unix terminal commands... Running python codes:

```
import os

class Wifi:

        def turn_off_wifi(self):
                cmd = ’sudo ifconfig wlan0 down’
                print(”turn off wifi”)
                os.system(cmd)

        def turn_on_wifi(self):
                cmd = ’sudo ifconfig wlan0 up’
                print(”turn on wifi”)
                os.system(cmd)

        def generate_wpa(self,SSID, password):
        os.system(”sudo chown pi /etc/wpa_supplicant/wpa_supplicant.conf”)
        #This is country, you should change it.
        #You can see your country code in this link
        #https://en.wikipedia.org/wiki/Country_code
                config_lines = [
                        ’ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev’
                        ’\nupdate_config=1’
                        ’\ncountry=TR’
                        ’\n’,
                        ’network={’,
                        ’\tssid=”{}”’.format(SSID),
                        ’\tpsk=”{}”’.format(password),
                        ’\tkey_mgmt=WPA-PSK’,
                        ’}’
                ]
                config = ’\n’.join(config_lines)

                with open(”/etc/wpa_supplicant/wpa_supplicant.conf”, ”w+”) as wifi:
                        wifi.write(config)

    def connect_wifi(self,ssid, password):
        #time.sleep(0.01)
        self.generate_wpa(ssid,password)
        #time.sleep(0.01)
        os.system(”sudo systemctl daemon-reload”)
        os.system(”sudo systemctl restart dhcpcd”)
```

4 Disable chromium update dialog on raspbian

In the file system under /etc/chromium-browser/customizations, open a new file named 01-disable-update-check:

```
cd /etc/chromium-browser/customizations
sudo vi 01-disable-update-check
```

Then type in the following single line of text:

```
CHROMIUM_FLAGS=”${CHROMIUM_FLAGS} --check-for-update-interval=31536000”
```

5 RPI Open a Chromium Web App Kiosk Mode

This is a Flask app that created via Python.

```
sudo nano /etc/xdg/lxsession/LXDE-pi/autostart
```

Add these lines:

```
@xscreensaver -no-splash
@xset s noblank
@xset s off
@xset -dpms
point-rpi
@sudo chmod 755 app.sh
@bash /home/pi/app.sh
@chromium-browser --kiosk --app=http://127.0.0.1:5000/
--incognito
--disable-translate
--noerrdialogs
--disable-session-crashed-bubble
--disable-infobars
```

Also we must create app.sh file:

```
#!/bin/bash
cd /home/pi/Desktop
python control.py
cd /home/pi/Desktop/teknotek
source venv/bin/activate
export FLASK_APP=app.py
flask run --host=0.0.0.0 --port=5000
```

6 RPI Cloning SD Card Use-Block

For this purpose we must each boot stage, we must control the CPU serial number includes to txt file that cretated previously via python.If it is not include then RPI shut down immediately.This python codes for it:

```
from subprocess import call
def getserial():
        # Extract serial from cpuinfo file
        cpuserial = "0000000000000000"
        try:
                f = open('/proc/cpuinfo','r')
                for line in f:
                        if line[0:6]=='Serial':
                                cpuserial = line[10:26]
                f.close()
        except:
                cpuserial = "ERROR000000000"

        return cpuserial

serial = getserial()

with open('/home/pi/Desktop/serials.txt') as f:
        try:
                if serial in f.read():
                        pass
                else:
                        print("kapanacak")
                        call("sudo␣shutdown␣-h␣now", shell=True)
        except:
                print("kapanacak")
                call("sudo␣shutdown␣-h␣now", shell=True)
```

We can access RPI CPU serial number with this terminal command:

```
cat /proc/cpuinfo | grep Serial | cut -d '␣' -f 2
```

And also we should open a file in Desktop named serials.txt:

```
touch /home/pi/Desktop/serials.txt
sudo nano /home/pi/Desktop/serials.txt
```

And we can add RPI serial number like 00000000f8f68405