Rust on Raspberry PI Zero

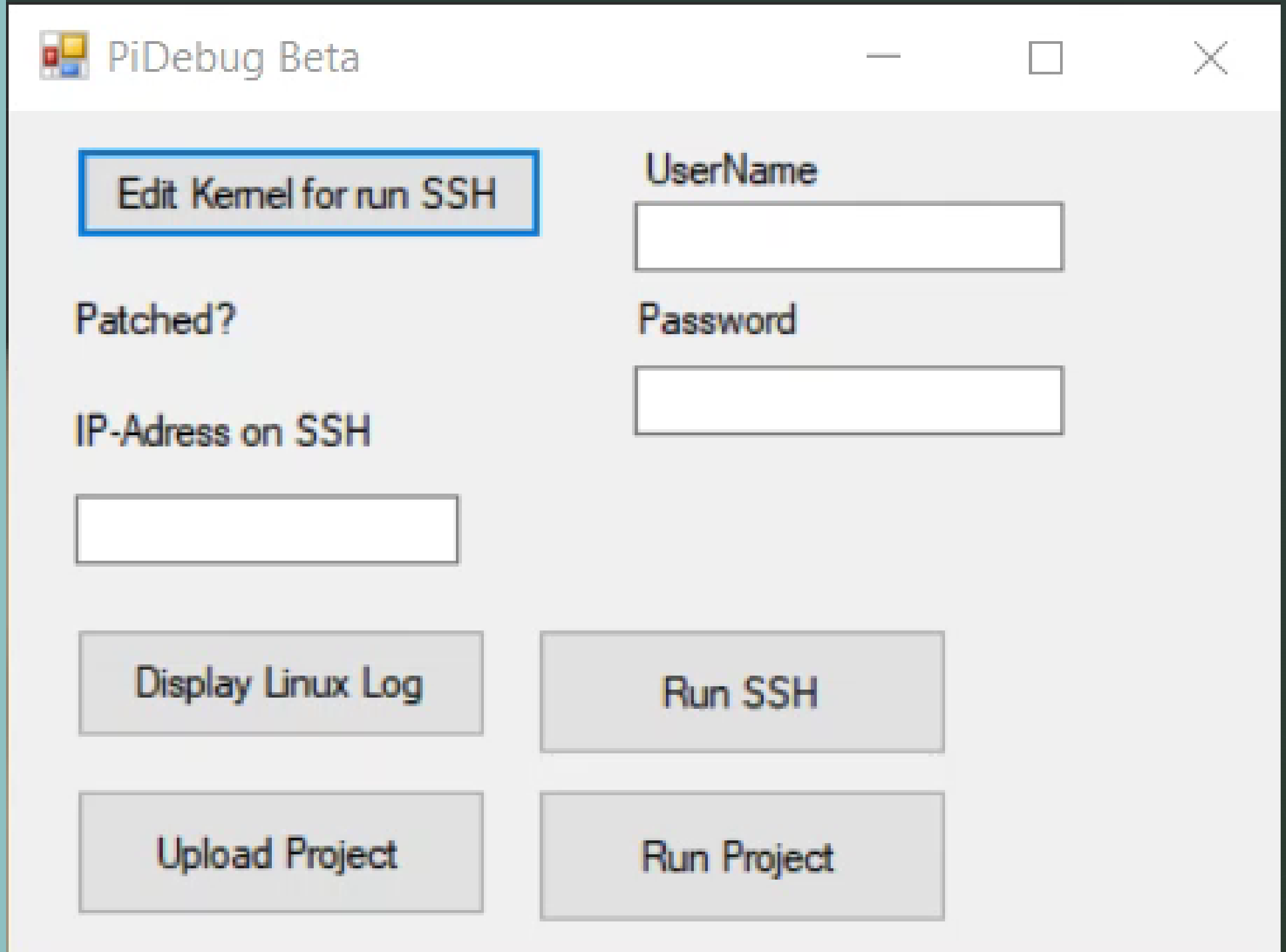
User Guide

Software Requirements:

- 1) Windows 10 with last updates (2004)
- 2) PiDebug V0.0.1 Beta
- 3) VirtualBox 6.12
- 4)Ubuntu / Ubuntu Mate 18.0.4.5 LTS
- 5)Rust (Last version)
- 6)Raspberry PI Image Flasher

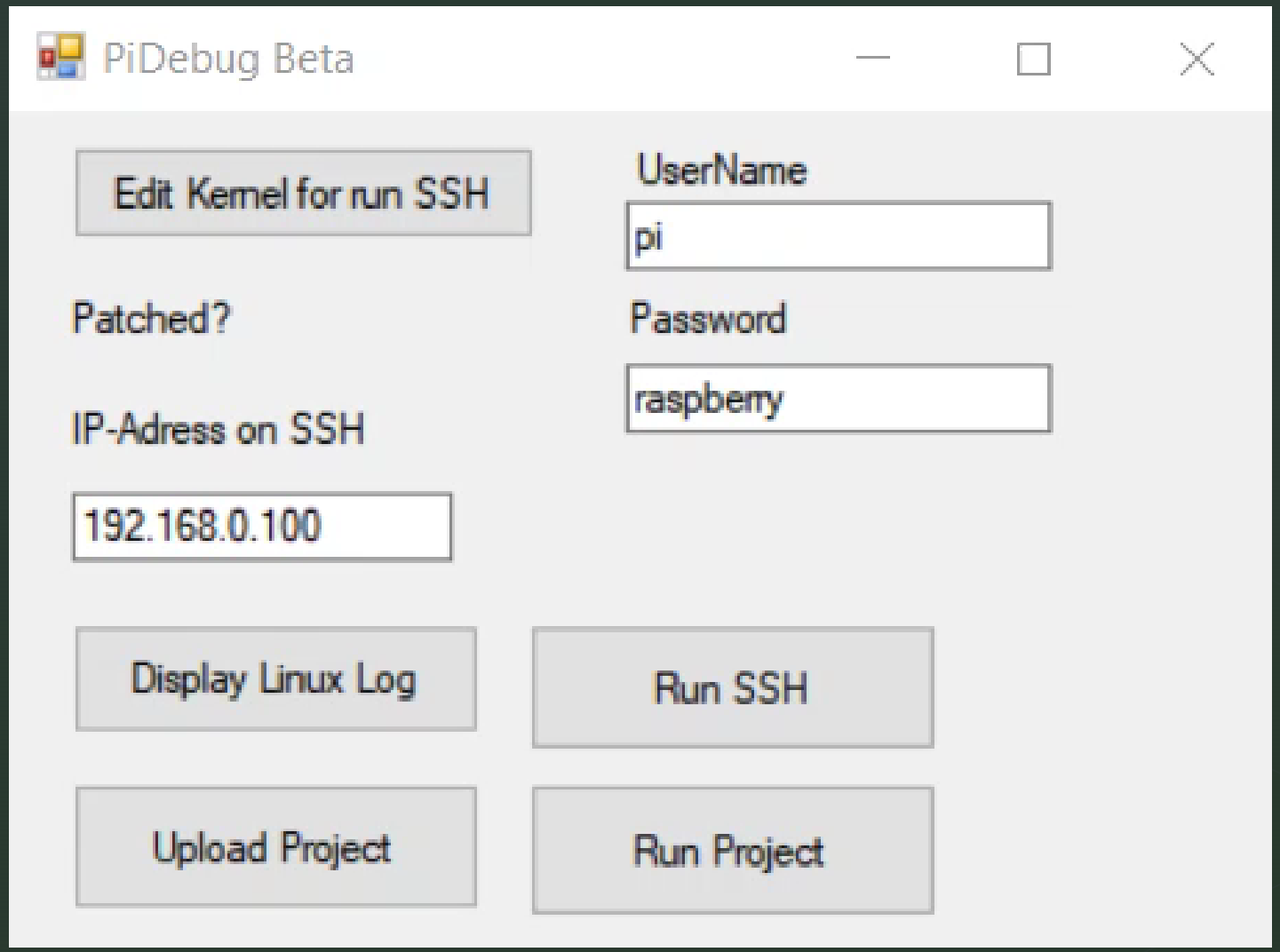
First Step (Prepare SSH)

- Flash SD-card via RPI Writter
- Open PiDebug and Press Edit Kernel Button.
- Select a boot volume and press OK
- Wait while text Patched ? : will changed and become olive green.


PiDebug Beta
Edit Kernel for run SSH
UserName
Patched?
Password
IP-Adress on SSH
Display Linux Log
Run SSH
Upload Project
Run Project

Next Install SD card to RPI and connect in to Router.

After it enter all creditals to PiDebug and check connection via getting Linux Log. After it connect via SSH.



Install&Configure Samba

- Install Samba by . `sudo apt-get install samba samba-common-bin`
- Create folder by : `sudo mkdir -m 1777 /share`
- Open Samba config by : `sudo nano /etc/samba/smb.conf`
- Add to bottom of file this config :

```
[share]
Comment = Pi shared folder
Path = /share
Browseable = yes
Writeable = Yes
only guest = no
create mask = 0777
directory mask = 0777
```

Reboot RPI after it

192.168.0.100 - PuTTY

```
root@raspberrypi:/home#
```

- Open Network Option in Explorer and try access to folder that Raspberry is sharing.

This PC

File | Computer | View

This PC

Search This PC

- Project
- project_tracker
- Screenshots
- welcome
- This PC
  - 3D Objects
  - Desktop
  - Documents
  - Downloads
  - Music
  - Pictures
  - Videos
  - Local Disk (C:)
- Network

Folders (7)

- 3D Objects
- Desktop
- Documents
- Downloads
- Music
- Pictures
- Videos

Devices and drives (1)

Local Disk (C:)

1.61 TB free of 1.81 TB

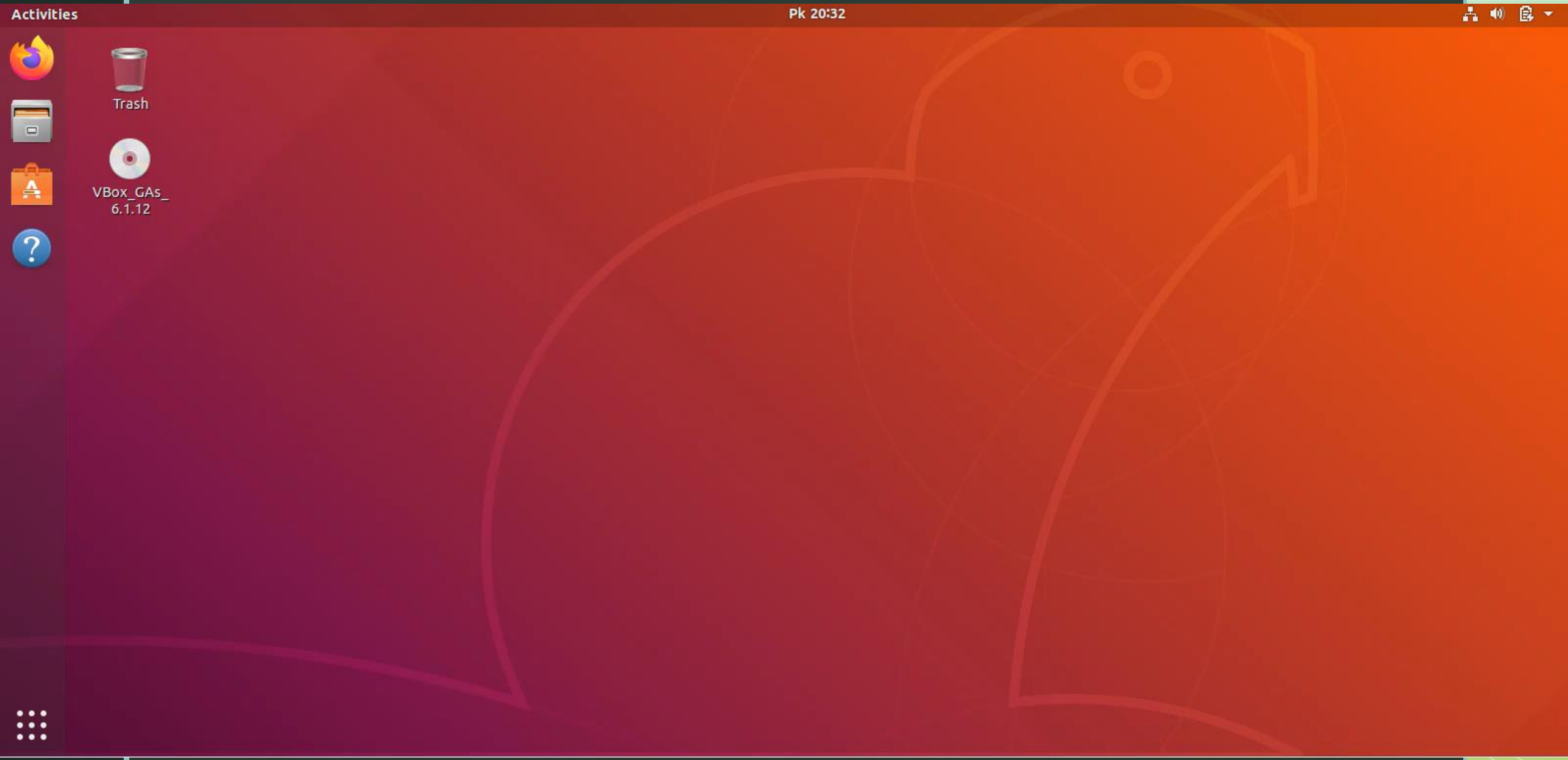
8 items

- Create Virtual Machine and install Ubuntu.

1) Make : sudo apt update
2) Install VirtualBox Additions
3) Install rustup using : curl https://sh.rustup.rs -sSf | sh
4) Set default toolchain by : rustup default stable
5) Install all dependencies by : sudo apt-get install gcc-arm-linux-gnueabihf libc6-armhf-cross libc6-dev-armhf-cross
6) Set another toolchain by : rustup target add arm-unknown-linux-gnueabihf
7) In user folder create new folder .cargo
8) Step into it
9) Create file config.toml
10) Open it with nano by : sudo nano config.toml

- File content :
- [target.arm-unknown-linux-gnueabihf]
- linker = "$HOME/rpi_tools/arm-bcm2708/arm-linux-gnueabihf/bin/arm-linux-gnueabihf-gcc"

1) Set one more toolchain by : rustup target add arm-unknown-linux-gnueabi
2) Get c++ linkers by : git clone https://github.com/raspberrypi/tools $HOME/rpi_tools
3) set it by : RUSTFLAGS="-C linker=$HOME/rpi_tools/arm-bcm2708/arm-rpi-4.9.3-linux-gnueabihf/bin/arm-linux-gnueabihf-gcc" cargo build --target arm-unknown-linux-gnueabihf --tests
4) Finally build project using: cargo build --target=arm-unknown-linux-gnueabihf


Activities
Pk 20:32
Trash
VBox_GAs_
6.1.12