

# GRUB rescue fix guide

parrot os + windows 11 dual boot

grub failed suddenly

i was deleting a partition on ssd. it worked fine till then but failed when i tried to copy file from hdd to hdd

grub rescue is only available

Got it — you've got **Parrot OS + Windows 11 dual boot**, and after deleting a partition on your SSD, **GRUB failed** and now you're stuck at the `grub rescue>` prompt. Let's fix it step-by-step.

## What Happened

When you deleted a partition, it's likely that:

- The **GRUB bootloader** (which was installed on that partition or referenced files there) lost access to its boot files ( `/boot/grub` ), OR
- The partition numbering changed, breaking GRUB's references.

## Step 1: Identify the Problem in `grub rescue`

At the `grub rescue>` prompt, try listing your drives and partitions:

```
bash
```

```
grub rescue> ls
```

You'll see something like:

```
scss
```

```
(hd0) (hd0,msdos1) (hd0,msdos2) (hd1,msdos1)
```

Now, test each to find your Linux `/boot` partition:

```
bash

grub rescue> ls (hd0,msdos1)/
```

Keep testing until one shows something like:

```
bash

/boot /etc /home ...
```

Note which one that is (e.g. `(hd0,msdos2)` ).

---

## Step 2: Set Root and Prefix

Once you find the correct partition (say it's `(hd0,msdos2)` ):

```
ruby

grub rescue> set root=(hd0,msdos2)
grub rescue> set prefix=(hd0,msdos2)/boot/grub
grub rescue> insmod normal
grub rescue> normal
```

If it works, GRUB will load normally and you'll get your boot menu.

If not, try other partitions or check if `/boot/grub` exists.

---

## Step 3: Boot into Parrot OS Live USB

If you can't boot into Parrot, use a **Parrot OS Live USB** (or any Debian-based live USB).

1. Boot from the USB.
2. Open a terminal.
3. Mount your Linux root partition:

```
bash

sudo mount /dev/sdXY /mnt
```

(Replace `sdXY` with your Parrot partition, e.g., `/dev/sda2`)

4. If you have a separate boot partition, mount it too:

```
bash

sudo mount /dev/sdXZ /mnt/boot
```

5. Mount essential filesystems:

```
bash

for i in /dev /proc /sys /run; do sudo mount --bind $i /mnt$i; done
```

6. Chroot into your system:

```
bash

sudo chroot /mnt
```

7. Reinstall GRUB:

```
bash

grub-install /dev/sdX
update-grub
```

(Replace `/dev/sdX` with your disk, not a partition — e.g., `/dev/sda`)

8. Exit and reboot:

```
bash

exit

sudo reboot
```



## Step 4: Ensure Windows Boot Option Exists

After rebooting, GRUB should detect Windows automatically.

If not, run again inside Parrot:

```
bash

sudo update-grub
```

You should see:

```
nginx
```

```
Found Windows Boot Manager on /dev/sdXY
```

## ⚠ If GRUB Still Fails

If after reinstalling, it still drops you into rescue mode:

- Check if the deleted partition was `/boot` or EFI ( `/boot/efi` ).
- For **UEFI systems**, ensure the EFI partition ( `/dev/sdXY` ) is mounted at `/boot/efi` before reinstalling GRUB:

```
bash
```

```
sudo mount /dev/sdXY /mnt/boot/efi
```

Would you like to tell me whether your system is **UEFI** or **Legacy (BIOS)**?

That changes the exact GRUB reinstall command slightly, and I can give you the exact ones for your setup.

↳ msdos1

do i need to type this mine shows gpt1 to 8

Perfect — that detail helps a lot. 👍

If your `ls` output in **grub rescue** shows partitions like this:

```
scss
```

```
(hd0,gpt1) (hd0,gpt2) (hd0,gpt3) ... (hd0,gpt8)
```

that means your disk uses **GPT partitioning** (modern UEFI-style layout).

So yes — everywhere you see `(hd0,msdos1)` in examples online, in your case it should be `(hd0,gpt1)` etc.



## Here's what to do

At the `grub rescue>` prompt:

### 1. List partitions:

```
bash

grub rescue> ls
```

### 2. Test each partition to see which contains your Linux filesystem:

```
bash

grub rescue> ls (hd0,gpt1)/
grub rescue> ls (hd0,gpt2)/
grub rescue> ls (hd0,gpt3)/
...
```

Look for one that shows folders like:

```
bash

/boot /etc /home /usr ...
```

That's your **Linux root** partition.

### 3. Once you find it (say it's `(hd0,gpt6)`), run:

```
bash

grub rescue> set root=(hd0,gpt6)
grub rescue> set prefix=(hd0,gpt6)/boot/grub
grub rescue> insmod normal
grub rescue> normal
```

→ This should bring you into the GRUB menu.

---

If it **still fails** with `error: file '/boot/grub/i386-pc/normal.mod' not found`, it means you're using **UEFI** and GRUB is looking in the wrong place.

To fix that, you'll need to boot from a **Parrot OS Live USB** and reinstall GRUB in **UEFI mode** (I can guide you step-by-step once you confirm).

---

### ✓ Next step for you:

Please tell me:

1. Does your system boot in **UEFI** or **Legacy/CSM** mode?
2. Do you have access to a **Parrot OS live USB**?

Once you confirm those, I'll give you *your exact commands* to reinstall GRUB cleanly without touching your data.

all filesystem are unknown

Got it — if `ls (hd0,gptX)` shows “unknown filesystem” for all partitions, that means GRUB can't read *any* of your partitions — which usually means one of these:

1. You deleted or damaged the partition that held `/boot` or `/boot/efi`.
2. GRUB modules for reading modern filesystems (ext4, btrfs, etc.) aren't loaded in rescue mode.
3. Your disk layout changed, and GRUB's internal references are outdated.

Don't worry — your data is *very likely still intact*, but GRUB can't see it from this minimal rescue shell.

Let's recover it properly. 🙌

---

## 🔧 Step 1: Boot from a Live USB

You'll need a **Parrot OS live USB** (or Debian/Ubuntu live USB — any will do).

1. Boot your computer from the live USB.
  - If your BIOS/UEFI doesn't boot it automatically, press **F12**, **Esc**, or **F9** at startup to choose the USB drive.
2. Once you're on the live desktop, open a **terminal**.

---

## 🧭 Step 2: Identify Your Linux and EFI Partitions

Run this to see all partitions:

```
bash
```

```
sudo fdisk -l
```

Look for:

- **Linux filesystem** → likely `ext4` (for Parrot)

- **EFI System Partition** → type `EFI` , formatted as `FAT32` , usually around 100–500 MB

Example output:

```
bash
```

```
Device    Type    Size
/dev/nvme0n1p1 EFI System 512M
/dev/nvme0n1p2 Microsoft reserved
/dev/nvme0n1p3 Windows
/dev/nvme0n1p6 Linux filesystem
```

Let's say your Parrot partition is `/dev/nvme0n1p6` and EFI is `/dev/nvme0n1p1` .

---

### ⚙️ Step 3: Mount and Reinstall GRUB (UEFI version)

Run these commands (replace with your actual partition names):

```
bash
```

```
sudo mount /dev/nvme0n1p6 /mnt
sudo mount /dev/nvme0n1p1 /mnt/boot/efi
for i in /dev /proc /sys /run; do sudo mount --bind $i /mnt$i; done
sudo chroot /mnt
```

Now you're "inside" your installed Parrot system.

Reinstall GRUB for UEFI:

```
bash
```

```
grub-install --target=x86_64-efi --efi-directory=/boot/efi --bootloader-id=Parrot
update-grub
```

Then exit and reboot:

```
bash
```

```
exit
sudo reboot
```

---

## After Reboot

You should see the GRUB menu again, with both **Parrot OS** and **Windows 11** listed.

If not, boot into Parrot and run:

```
bash  
  
sudo update-grub
```

If you'd like, you can paste the output of:

```
bash  
  
sudo fdisk -l
```

from your live USB here — I can tell you *exactly* which partitions to mount and what commands to use (so there's no risk of overwriting anything).


Would you like to do that?

windows boot manager works fine only issue is with  
linux bootloader grub

Perfect — that's actually *good news!* 

If **Windows Boot Manager works fine**, then your EFI system and partitions are intact — only **GRUB's bootloader files** are missing or broken.

That means recovery is straightforward — you just need to **reinstall GRUB into the EFI partition**.

Let's fix it cleanly. 

## What You'll Need

- A **Parrot OS live USB** (or any Debian-based live ISO — Ubuntu, Kali, etc. also works).
- Internet not required.
- About 5 minutes.



## Step-by-Step Fix (UEFI System)

### 1. Boot into your Parrot OS live USB

Choose “Try Parrot OS” (not install).

Once you’re at the desktop, open a **terminal**.

---

### 2. Identify your partitions

Run:

```
bash

sudo fdisk -l
```

Look for:

- Your **Linux root** partition — likely labeled as `Linux filesystem`, type `ext4`.  
→ Example: `/dev/nvme0n1p6` or `/dev/sda6`
- Your **EFI System Partition (ESP)** — type “EFI System”, `FAT32`, around 100–500 MB.  
→ Example: `/dev/nvme0n1p1` or `/dev/sda1`

Take note of both.

---

### 3. Mount the partitions

Replace the device names below with yours:

```
bash

sudo mount /dev/nvme0n1p6 /mnt
sudo mount /dev/nvme0n1p1 /mnt/boot/efi
for i in /dev /proc /sys /run; do sudo mount --bind $i /mnt$i; done
```

### 4. Chroot into your installed system

```
bash
```

```
sudo chroot /mnt
```

You're now "inside" your real Parrot installation.

---

## 5. Reinstall GRUB

Run this:

```
bash

grub-install --target=x86_64-efi --efi-directory=/boot/efi --bootloader-id=Parrot
update-grub
```

You should see messages like:

```
nginx

Installing for x86_64-efi platform.
Installation finished. No error reported.
Found Windows Boot Manager on /dev/nvme0n1p3
```

---

## 6. Exit and reboot

```
bash

exit

sudo reboot
```

After reboot, you should see the **GRUB menu** again, with both Parrot OS and Windows 11 listed. 🎉

---

### ✓ Optional: Set GRUB as Default Boot Entry

If Windows still boots by default, run this (inside Parrot):

```
bash

sudo efibootmgr
```

Find the entry for `Parrot` and note its Boot number (e.g. `Boot0003` ), then:

```
bash

sudo efibootmgr -o 0003,0001
```

(Adjust the numbers — `0003` should be your Parrot entry, `0001` your Windows entry.)

---

If you'd like, you can paste the output of:

```
bash

sudo fdisk -l
```

and I'll tell you *exactly* which devices to mount so you don't risk mounting the wrong one.

Would you like me to guide you through that with your real partition layout?

i have opened parrot home live boot menu 6.3.2

Perfect 🔥 — you're in **Parrot Home Live 6.3.2**, exactly what we need.

Let's now fix your GRUB **step by step safely**.

Follow these commands **carefully** — I'll explain each as we go.

---

## Step 1. Open a Terminal

Once on the Parrot desktop, open the **Terminal** (the black icon on the top bar or from Applications → System → Terminal).

---

## Step 2. List all partitions

Run:

```
bash

sudo fdisk -l
```

You'll see something like:

```
bash
```

```
Disk /dev/nvme0n1: 512GB
```

```
Device      Start   End Sectors Size Type
```

```
/dev/nvme0n1p1 2048 1230847 1228800 600M EFI System
```

```
/dev/nvme0n1p2 ...      ...      ... Microsoft reserved
```

```
/dev/nvme0n1p3 ...      ...      ... Microsoft basic data
```

```
/dev/nvme0n1p6 ...      ...      ... Linux filesystem
```

Now tell me what it shows — or just **copy and paste that output here**.

I'll tell you *exactly* which ones are:

- your **Linux root partition** (where Parrot is installed)
- your **EFI System Partition** (where GRUB needs to go)

Then I'll give you the exact mount + grub repair commands for your setup.

---

Once you share that output, we can finish the repair in 3–4 commands.

↳ \* Your **Linux root** partition — likely labeled as `Linux filesystem`, type `ext4`. →  
Example: `/dev/nvme0n1p6` or `/dev/sda6` \* Your **EFI System Partition (ESP)** — type  
"EFI System", `FAT32`, around 100–500 MB. → Example: `/dev/nvme0n1p1` or `/dev/sda1`

```
/dev/nvme0n1p1 (EFI SYSTEM)
/dev/nvme0n1p7 (LINUX FILESYSTEM)
```

Perfect — thank you! ✓

So you have:

- **EFI partition:** `/dev/nvme0n1p1`
- **Linux (Parrot) root partition:** `/dev/nvme0n1p7`

Now we can safely reinstall GRUB.

---

## ⚙️ Step-by-Step Commands

Please copy and paste these **exactly** in your terminal one by one 🙌

### 1 Mount your Linux and EFI partitions

```
bash
```

```
sudo mount /dev/nvme0n1p7 /mnt  
sudo mount /dev/nvme0n1p1 /mnt/boot/efi  
for i in /dev /proc /sys /run; do sudo mount --bind $i /mnt$i; done
```

---

## 2 Enter your installed Parrot system

```
bash  
  
sudo chroot /mnt
```

---

## 3 Reinstall GRUB for UEFI

```
bash  
  
grub-install --target=x86_64-efi --efi-directory=/boot/efi --bootloader-id=Parrot  
update-grub
```

You should see messages like:

```
nginx  
  
Installing for x86_64-efi platform.  
Installation finished. No error reported.  
Found Windows Boot Manager on /dev/nvme0n1p3
```

---

## 4 Exit and reboot

```
bash  
  
exit  
sudo reboot
```

---

After reboot, you should see the **GRUB menu** again, showing both **Parrot OS** and **Windows Boot Manager** 🎉

If your computer still boots straight into Windows, I'll show you how to make Parrot the default in the EFI boot order — but let's first confirm if GRUB appears after the reboot.

Can you try these commands and tell me what happens after the restart?

↳ ``sudo chroot /mnt``

CHROOT: FAILED TO RUN COMMAND  
/BIN/BASH NO SUCH FILE OR DIR

Good — that error tells us something very specific:

your **Parrot root partition mounted correctly**, but it's missing or not finding

`/bin/bash` — usually because either:

- The partition didn't mount properly (wrong path or missing subvolume if Btrfs),  
or
- `/mnt` is empty because Parrot is installed somewhere else (e.g., encrypted or nested inside `/dev/mapper/...`).

Let's confirm what's really mounted.

---

## Step 1: Check if your files are there

Run:

```
bash
```

```
ls /mnt
```

You should see something like:

```
arduino
```

```
bin boot etc home lib usr var ...
```

If you **don't** see those (if it's empty or only has "lost+found"), it means

`/dev/nvme0n1p7` isn't your real root partition — or it's encrypted (LUKS).

Please tell me what `ls /mnt` shows.

---

## Step 2: If `/mnt` is empty, check partitions again

Run:

```
bash

sudo fdisk -l
lsblk -f
```

Look for something like `/dev/mapper/parrot--vg-root` or a partition with `crypto_LUKS` or `LVM2_member` under “FSTYPE”.

If that’s there, your Parrot install uses **encryption or LVM**, so we need to unlock/mount it first.

---

Once you share either:

- the result of `ls /mnt`, or
- the output of `lsblk -f`,

I’ll tell you **exactly** what to mount so `chroot` works this time.

↳ ``bin boot etc home lib usr var ...``

I ONLY SEE @ BOOT @HOME PROC RUN SYS

Perfect — that’s actually a *huge clue*.

Seeing only

```
arduino

@ boot @home proc run sys
```

means your Parrot partition uses **Btrfs with subvolumes** — the `@` and `@home` are Btrfs subvolumes for root (`/`) and `/home`.

So GRUB can’t find `/bin/bash` because you mounted the *Btrfs filesystem* but not the **root subvolume (@)**.

Let’s fix that — very easy now. 🙌

---

## Step-by-Step (for Btrfs system)

### 1 Unmount everything from before

Run these to clean up:

```
bash
```

```
sudo umount -R /mnt
```

---

### 2 Mount your root subvolume correctly

```
bash
```

```
sudo mount -o subvol=@ /dev/nvme0n1p7 /mnt
```

---

### 3 Mount EFI and system directories again

```
bash
```

```
sudo mount /dev/nvme0n1p1 /mnt/boot/efi
```

```
for i in /dev /proc /sys /run; do sudo mount --bind $i /mnt$i; done
```

---

### 4 Try chroot again

```
bash
```

```
sudo chroot /mnt
```

✓ This time it should drop you into your Parrot system (you'll see your username or `root@parrot` prompt).

---

### 5 Reinstall GRUB (UEFI)

Once inside:

```
bash
```



```
grub-install --target=x86_64-efi --efi-directory=/boot/efi --bootloader-id=Parrot  
update-grub
```

---

## 6 Exit and reboot

```
bash  
  
exit  
sudo reboot
```

---

After reboot, you should see your **GRUB menu** again with both Parrot and Windows entries. 🎉

---

Would you like me to also show how to set GRUB as the default boot option (in case Windows Boot Manager keeps taking over)?