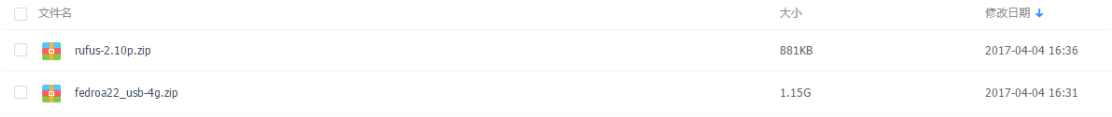
USB Drive Maintenance System Introduction

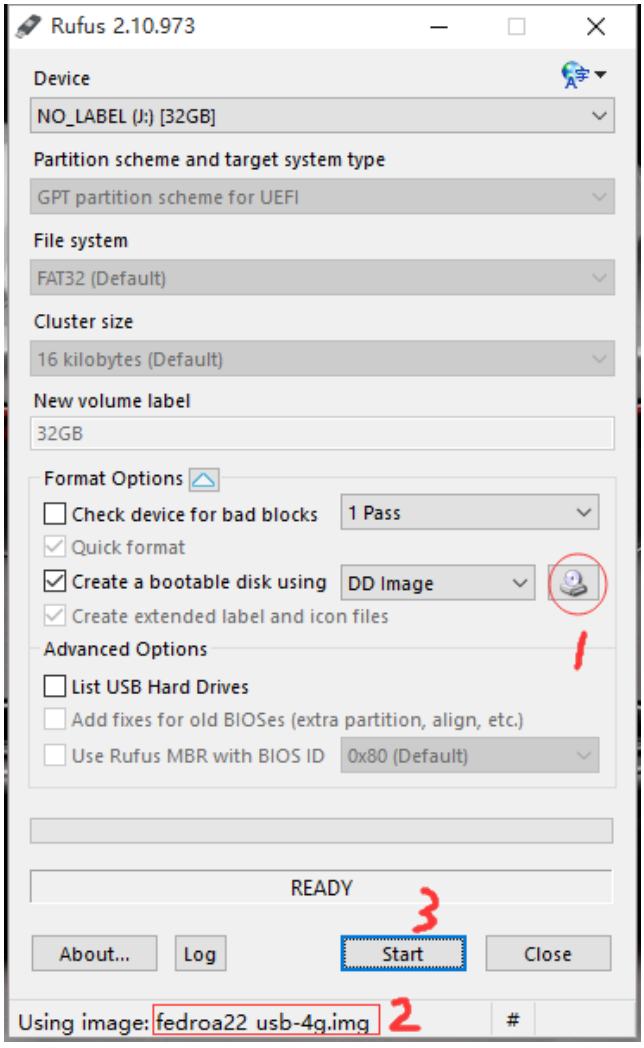
Step 1:

Download the files we need from Baidu Cloud.



“rufus-2.10p.zip” is a tool to create bootable USB drives and “fedroa22_usb-4g.zip” is a USB live fedora system image file. Unzip the fedroa22_usb-4g.zip file, then you will get a “fedroa22_usb-4g.img” file. Insert the USB drive, then open rufus, choose the “.img” file, after that write the file into USB drive.

Now you have a bootable USB drive maintenance system.



Step 2:

Insert the bootable USB drive into UC300, It will boot from the USB drive and enter the maintenance system automatically. When you heard “beep”, that means the system has been loaded. The SSH service is running by default, you can login and operate the system by putty.

IP：**172.16.101.1**

Username: **root**

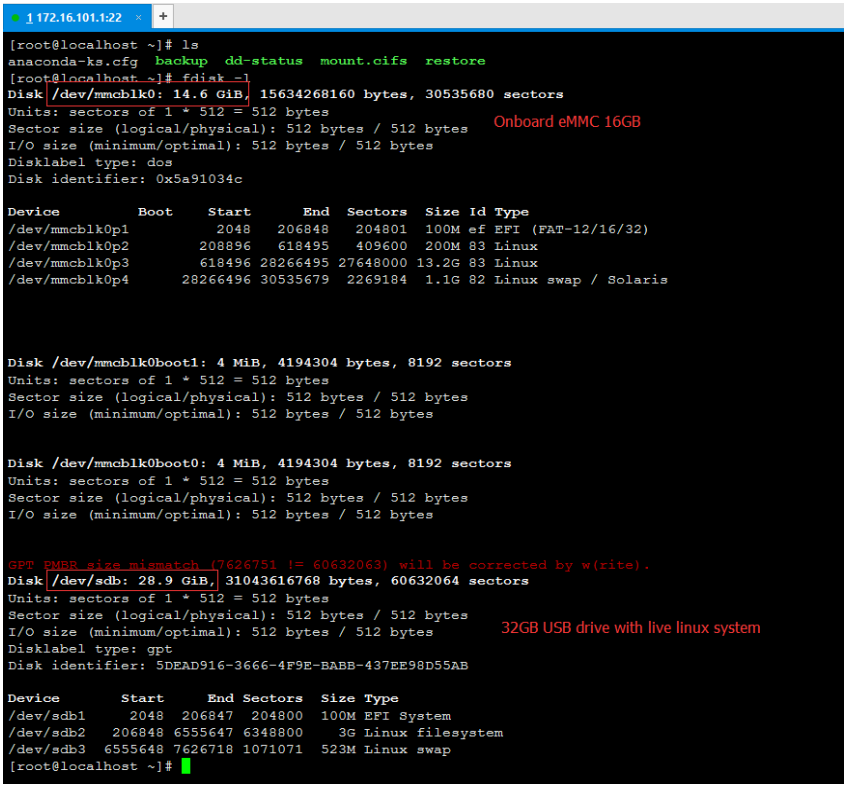
Password：**111111**

Step 3:

If your USB drive capacity is 32G or more, you could put the downloaded UC300 system image into USB drive, and write into the onboard eMMC flash from USB drive.

Step 4:

Take 32G USB drive as an example to introduce the steps of building partitions and mount.



1. Divide free space of USB drive into the 4th partition with fdisk.

```
[root@localhost ~]# fdisk /dev/sdb

Welcome to fdisk (util-linux 2.26.2).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

GPT PMBR size mismatch (7626751 != 60632063) will be corrected by w(rite).
GPT PMBR size mismatch (7626751 != 60632063) will be corrected by w(rite).

Command (m for help): p

Disk /dev/sdb: 28.9 GiB, 31043616768 bytes, 60632064 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 5DEAD916-3666-4F9E-BABB-437EE98D55AB

Device       Start     End Sectors  Size Type
/dev/sdb1     2048  206847  204800  100M EFI System
/dev/sdb2   206848 6555647 6348800    3G Linux filesystem
/dev/sdb3  6555648 7626718 1071071  523M Linux swap

Command (m for help): n
Partition number (4-128, default 4):
First sector (7626719-60632030, default 7626752):
Last sector, +sectors or +size{K,M,G,T,P} (7626752-60632030, default 60632030):

Created a new partition 4 of type 'Linux filesystem' and of size 25.3 GiB.

Command (m for help): p
Disk /dev/sdb: 28.9 GiB, 31043616768 bytes, 60632064 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 5DEAD916-3666-4F9E-BABB-437EE98D55AB

Device       Start      End  Sectors  Size Type
/dev/sdb1     2048   206847   204800  100M EFI System
/dev/sdb2   206848  6555647  6348800    3G Linux filesystem
/dev/sdb3  6555648  7626718  1071071  523M Linux swap
/dev/sdb4  7626752 60632030 53005279 25.3G Linux filesystem

Command (m for help): w
GPT PMBR size mismatch (7626751 != 60632063) will be corrected by w(rite).
```

After finished partitions, reboot the system.

2. Login the USB drive system again and format the 4th partition as ext4.

```
[root@localhost ~]# mkfs.ext4 /dev/sdb4
mke2fs 1.42.12 (29-Aug-2014)
Creating filesystem with 6625659 4k blocks and 1656480 inodes
Filesystem UUID: 974d62fe-0b06-4535-9b4a-357b7d83dc9a
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

[root@localhost ~]# blkid
/dev/mmcblk0: PTUUID="5a91034c" PTTYPE="dos"
/dev/mmcblk0p1: SEC_TYPE="msdos" UUID="F1FF-1792" TYPE="vfat" PARTUUID="5a91034c-01"
/dev/mmcblk0p2: UUID="92ec9203-8a2e-4103-9e8b-f6955378443a" TYPE="ext4" PARTUUID="5a91034c-02"
/dev/mmcblk0p3: UUID="1ceda188-3251-4908-ba11-5aefab2dedd2" TYPE="ext4" PARTUUID="5a91034c-03"
/dev/mmcblk0p4: UUID="04f7e088-8521-4ebc-b902-25c7e442b80b" TYPE="swap" PARTUUID="5a91034c-04"
/dev/sdb1: SEC_TYPE="msdos" UUID="AF17-7585" TYPE="vfat" PARTUUID="b5697c30-9ac5-490f-bf6b-7ba15f76429c"
/dev/sdb2: UUID="c90b9e71-184f-4182-8c2a-e4d6afa9a7e7" TYPE="ext4" PARTUUID="00788f3a-2ce5-4639-8423-2b76f980b15e"
/dev/sdb3: UUID="30757543-3e07-4540-bfbd-43fe5d6cf6f7" TYPE="swap" PARTUUID="5cbc47f5-4732-48ce-8112-a9c7a23c6d68"
/dev/sdb4: UUID="974d62fe-0b06-4535-9b4a-357b7d83dc9a" TYPE="ext4" PARTUUID="a2e1aa60-d69b-4527-87ad-3af8a75576a6"
[root@localhost ~]# vi /etc/fstab
```

3. Search the UUID in the 4th partition by blkid command, and add it to the file "/etc/fstab". Then the 4th partition will be mounted to /mnt after starting up.

```
#
# /etc/fstab
# Created by anaconda on Wed Feb  3 22:52:39 2016
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=c90b9e71-184f-4182-8c2a-e4d6afa9a7e7 /                       ext4    defaults        1 1
UUID=974d62fe-0b06-4535-9b4a-357b7d83dc9a /mnt                    ext4    defaults        1 1
UUID=AF17-7585          /boot/efi               vfat    umask=0077,shortname=winnt 0 0
UUID=30757543-3e07-4540-bfbd-43fe5d6cf6f7 swap                    swap    defaults        0 0
~
```

4. Unzip the UC300 system image file, and move the ".img" file to the 4th partition (/mnt), rename it to "backup.img".

```
[root@localhost ~]# ls
anaconda-ks.cfg  backup  dd-status  mount.cifs  restore
[root@localhost ~]# cat backup
#!/bin/bash

dd if=/dev/mmcblk0 of=/mnt/backup.img bs=512K
[root@localhost ~]# cat restore
#!/bin/bash

dd if=/mnt/backup.img of=/dev/mmcblk0 bs=512K
[root@localhost ~]#
```

5. Execute the script "restore" above, wait about 15 minutes to finish writing the system image into eMMC flash.

   If you want to know the progress, you could open a new window and execute script "dd-status". Then back to the "restore" Window, you will get process status.

6. Once finished, execute command "poweroff", then remove the USB drive.

   Start up again, you could enter the new system now.

Enjoy!

Tips:

If your USB drive capacity is less than 32G, then you could choose a USB drive with capacity more than 4G. After writing the maintenance system image to USB drive and booting from USB drive, upload the UC300 system iamge file to a windows share storage. Then mount it to “/mnt”, then execute script “restore” to finish writing system into eMMC flash.

```
[root@localhost ~]# ls
anaconda-ks.cfg  backup  dd-status  mount.cifs  restore
[root@localhost ~]# cat mount.cifs
#!/bin/bash

mount.cifs -o rw,uid=0,gid=0,username=username,password=password //172.16.0.x/images  /mnt
[root@localhost ~]#
[root@localhost ~]# cat restore
#!/bin/bash

dd if=/mnt/backup.img of=/dev/mmcblk0 bs=512K
[root@localhost ~]#
[root@localhost ~]#
```

The script “mount.cifs” above is used for mounting windows shared storage, you just need to replace the IP address and user name/password.

```
[root@localhost ~]# ./mount.cifs
[root@localhost ~]# ./restore
[root@localhost ~]#
```