Linux

1.arch linux安装

软件环境

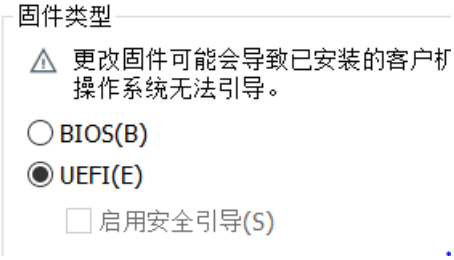
Windows版本: 21H2
VMware版本: Workstation Pro 16.2.4

硬件环境

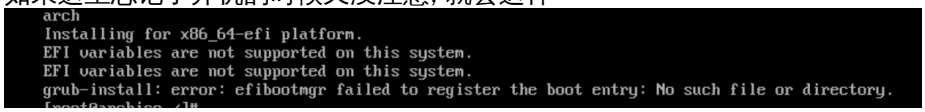
cpu: intel Core i5-7300HQ
gpu: Nvidia GTX1050
内存: 8GB
硬盘: 128M+1T

安装流程

从官网下载iso文件, 打开VMware, 创建新虚拟机, 这里有一个虚拟磁盘格式的选项,
如果虚拟机保存的磁盘和我一样是机械硬盘的话, 这个选项不要选高端的nvme,
否则在装好系统重启的时候会打不开磁盘
这里如果是在之前选择了保存在一个文件内, 可以考虑从这个文件新建一个虚拟机,
如果是保存在了多个文件内, 那大概是没救了
在开机前要进设置的高级, 把固件类型由BIOS改成UEFI



如果这里忘记了开机的时候又没注意, 就会这样



正式开机, 首先对硬盘进行分区, 我分了三个区, boot一个, swap一个

```
Command (m for help): g
Created a new GPT disklabel (GUID: 3ZF123E7-1E97-014D-89D5-273A9753AFB7).

Command (m for help): n
Partition number (1-128, default 1):
First sector (2048-67108830, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-67108830, default 67106815): +512M

Created a new partition 1 of type 'Linux filesystem' and of size 512 MiB.

Command (m for help): n
Partition number (2-128, default 2):
First sector (1050624-67108830, default 1050624):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (1050624-67108830, default 67106815): +2GB

Created a new partition 2 of type 'Linux filesystem' and of size 1.9 GiB.

Command (m for help): n
Partition number (3-128, default 3):
First sector (4956160-67108830, default 4956160):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (4956160-67108830, default 67106815):

Created a new partition 3 of type 'Linux filesystem' and of size 29.6 GiB.

Command (m for help): t
Partition number (1-3, default 3): 1
Partition type or alias (type L to list all): 1

Changed type of partition 'Linux filesystem' to 'EFI System'.

Command (m for help): t
Partition number (1-3, default 3): 2
Partition type or alias (type L to list all): 19

Changed type of partition 'Linux filesystem' to 'Linux swap'.

Command (m for help):
```

然后对分区进行格式化

```
root@archiso ~ # mkfs.btrfs /dev/sda3
btrfs-progs v6.0
See http://btrfs.wiki.kernel.org for more information.

NOTE: several default settings have changed in version 5.15, please make sure
      this does not affect your deployments:
      - DUP for metadata (-m dup)
      - enabled no-holes (-O no-holes)
      - enabled free-space-tree (-R free-space-tree)

Label:              (null)
UUID:               9fb15ffc-cce6-4aeb-b13d-8cff5ae4e388
Node size:          16384
Sector size:        4096
Filesystem size:    29.64GiB
Block group profiles:
  Data:             single            8.00MiB
  Metadata:         DUP             256.00MiB
  System:           DUP               8.00MiB
SSD detected:       no
Zoned device:       no
Incompat features:  extref, skinny-metadata, no-holes
Runtime features:   free-space-tree
Checksum:           crc32c
Number of devices:  1
Devices:
   ID        SIZE  PATH
    1    29.64GiB  /dev/sda3

root@archiso ~ # mkfs.fat -F32 /dev/sda1
mkfs.fat 4.2 (2021-01-31)
root@archiso ~ # mkswap /dev/sda2
Setting up swapspace version 1, size = 1.9 GiB (1999630336 bytes)
no label, UUID=47bdb572-5b66-4db0-843f-7ba4600ab3ae
```

创建子卷

```
root@archiso ~ # mount /dev/sda3 /mnt
root@archiso ~ # btrfs su cr /mnt/@
Create subvolume '/mnt/@'
root@archiso ~ # btrfs su cr /mnt/@home
Create subvolume '/mnt/@home'
root@archiso ~ # umount
umount: bad usage
Try 'umount --help' for more information.
1 root@archiso ~ # umount /mnt
```

正式挂载并打开透明压缩，打开swap分区

```
root@archiso ~ # mount -o compress=zstd,subvol=@ /dev/sda3 /mnt
root@archiso ~ # mkdir /mnt/home
root@archiso ~ # mount -o compress=zstd,subvol=@home /dev/sda3 /mnt/home
root@archiso ~ # mkdir /mnt/boot
root@archiso ~ # mount /dev/sda1 /mnt/boot
root@archiso ~ # swapon /dev/sda2
```

下载安装一些必要的软件包和linux内核，出于个人习惯一起下了vim，而且为了装好系统之后能上网下了dhcpcd

```
root@archiso ~ # pacstrap /mnt linux linux-firmware base base-devel vim dhcpcd_
```

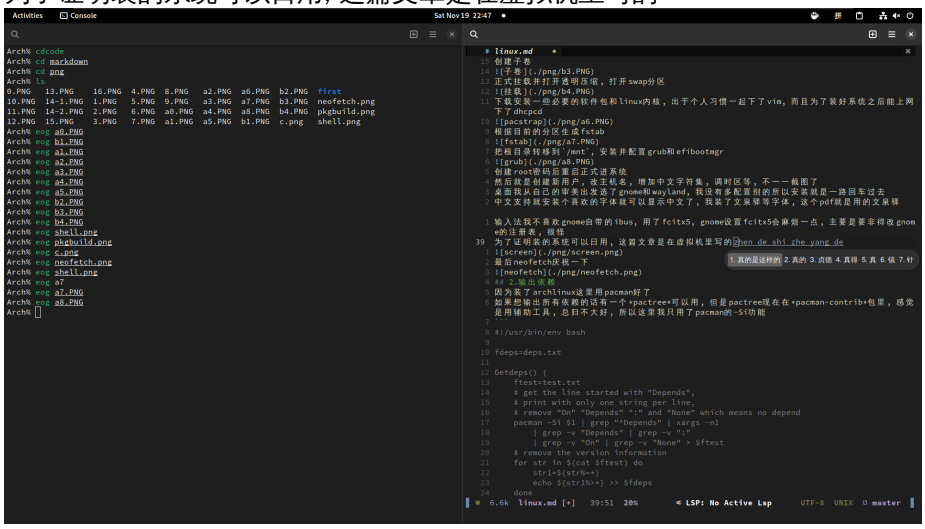
根据目前的分区生成fstab

```
root@archiso ~ # genfstab -U /mnt > /mnt/etc/fstab
root@archiso ~ # cat /mnt/etc/fstab
# /dev/nvme0n1p3
UUID=b3f0b767-40db-4d7d-b191-df049593de1a       /               btrfs           rw,relatime,compress=zstd:3,ssd,space_cache=v2,s
ubvolid=256,subvol=/@   0 1

# /dev/nvme0n1p1
UUID=30C3-31CC          /boot           vfat            rw,relatime,fmask=0022,dmask=0022,codepage=437,iocharset=ascii,shortname
=mixed,utf8,errors=remount-ro   0 2

# /dev/nvme0n1p3
UUID=b3f0b767-40db-4d7d-b191-df049593de1a       /home           btrfs           rw,relatime,compress=zstd:3,ssd,space_cache=v2,s
ubvolid=257,subvol=/@home       0 2

# /dev/nvme0n1p2
UUID=6540ccd5-2dd4-47f6-9630-89a02a52aa97       none            swap            defaults        0 0
```

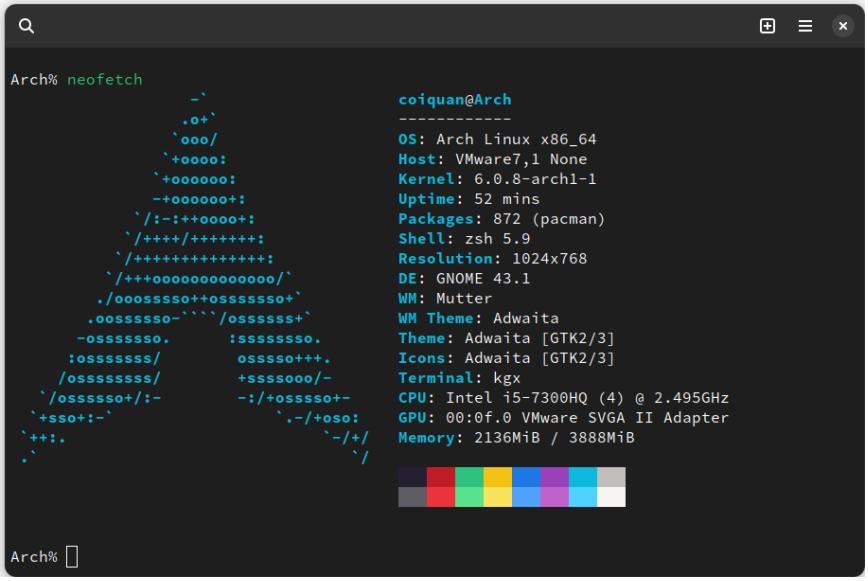
把根目录转移到/mnt，安装并配置grub和efibootmgr

```
[root@archiso /]# grub-install --target=x86_64-efi --efi-directory=/boot --bootloader-id=archcq
Installing for x86_64-efi platform.
Installation finished. No error reported.
[root@archiso /]# grub-mkconfig -o /boot/grub/grib.cfg
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-linux
Found initrd image: /boot/initramfs-linux.img
Found fallback initrd image(s) in /boot:  initramfs-linux-fallback.img
Warning: os-prober will not be executed to detect other bootable partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
Adding boot menu entry for UEFI Firmware Settings ...
done
```

创建root密码后重启正式进系统
然后就是创建新用户，改主机名，增加中文字符集，调时区等，不一一截图了
桌面我从自己的审美出发选了gnome和wayland，我没有多配置别的所以安装就是一路回车过去
中文支持就安装个喜欢的字体就可以显示中文了，　　我装了文泉驿等字体，
这个pdf就是用的文泉驿
输入法我不喜欢gnome自带的ibus，　用了fcitx5，　gnome设置fcitx5会麻烦一点，
主要是要非得改gnome的注册表，很怪
为了证明装的系统可以日用，这篇文章是在虚拟机里写的



最后neofetch庆祝一下



2.输出依赖

因为装了archlinux这里用pacman好了

如果想输出所有依赖的话有一个pactree可以用，　但是pactree现在在pacman-contrib包里．　感觉是用辅助工具，总归不大好，所以这里我只用了pacman的-Si功能

```
#!/usr/bin/env bash

fdeps=deps.txt

Getdeps() {
    ftest=test.txt
    # get the line started with "Depends",
    # print with only one string per line,
    # remove "On" "Depends" ":" and "None" which means no depend
    pacman -Si $1 | grep "^Depends" | xargs -n1
        | grep -v "Depends" | grep -v ":"
        | grep -v "On" | grep -v "None" > $ftest
    # remove the version information
    for str in $(cat $ftest) do
        str1=${str%=*}
        echo ${str1%>*} >> $fdeps
    done
    rm $ftest
}
```

```
fining=ining.txt
fined=ined.txt
echo $1 > $fining
# remove blank lines, break when the amount of lines is 0
while [ $(sed /^$/d $fining | wc -l ) -gt 0 ]; do
    echo > $fdeps
    for str in $(cat $fining); do
        echo $str >> $fined
        # redirect stderr to null,
        # because some packages can not found by pacman
        Getdeps $str 2> /dev/null
    done
     # print str only in deps to ining for the next loop
    comm -23 <(sort -u $fdeps)  <(sort -u $fined) > $fining
done
echo -e "Name: \c"
cat $fined
rm $fining $fined $fdeps
```

使用例

```
Arch% ./getdeps.sh clang
Name: clang
compiler-rt
gcc
llvm-libs
binutils
gcc-libs
libedit
libffi
libisl.so
libmpc
libxml2
ncurses
zlib
zstd
glibc
icu
jansson
libelf
libncursesw.so
lz4
mpfr
readline
xz
filesystem
gmp
libbz2.so
libcurl.so
linux-api-headers
sh
tzdata
iana-etc
Arch%
```

4.打包vim

这里还是用pacman
pacman打包只要写一个PKGBUILD就行，代码的下载和构建都会在makepkg的时候自动完成，我写的PKGBUILD是这样的

```
#Maintainer: Naidesu <test@test.test>
pkgname='vimtest'
pkgver=9.0.0905
pkgrel=1
pkgdesc='a simple pkgtest'
arch=('any')
url="naidesu.test"
license=('Vimlicense')
depends=('vim-runtime=9.0.0814-1' 'gpm' 'acl' 'glibc' 'libgcrypt' 'zlib' 'perl')
conflicts=('vim')
source=('https://github.com/vim/vim/archive/refs/tags/v9.0.0905.tar.gz')
md5sums=('SKIP')

build() {
    cd "${srcdir}"
    tar -zxvf "v${pkgver}.tar.gz"
    cd "vim-${pkgver}"
    make
}
package() {
    cd "${srcdir}/vim-${pkgver}"
    make DESTDIR=${pkgdir} install
}
```

然后执行makepkg，就会获得一个包，像这个样子

```
Arch% ls
pkg  PKGBUILD  src  v9.0.0905.tar.gz  vimtest-9.0.0905-1-any.pkg.tar.zst
Arch%
```

用`pacman -U`安装即可
~~当然因为大家肯定都装着vim，所以没办法安装成功的~~

5.编译内核

从官网下最新版本的内核下来解压
用`zcat /proc/config.gz > .config`获得一个和现在的内核一样的配置文件
阅读MakeFile可以发现如果装了clang就会优先用clang构建而不是gcc，所以没有改的必要，可以直接make
make运行了一会报错，没有bc命令，装上bc，再make
不久又报错了，`kernel/kheaders_data.tar.xz error 127`，查找某不知道名字的搜索引擎，装上cpio，再make
经过了漫长的等待之后，内核编译好了，然后再用`sudo make modules_install`来编译模块

把编译内核生成的bzImage文件复制到boot文件夹里， 改名叫vmlinuz-linuxtest,
虽然不太符合内核命名规则 然后cd进/etc/mkinitcpio.d/把linux.preset复制一份linuxtest.preset,
把文件内容改成这样

```
# mkinitcpio preset file for the 'linux' package

ALL_config="/etc/mkinitcpio.conf"
ALL_kver="boot/vmlinuz-linuxtest"

PRESETS=('default' 'fallback')

#default_config="/etc/mkinitcpio.conf"
default_image="boot/initramfs-linuxtest.img"
#default_options=""

#fallback_config="/etc/mkinitcpio.conf"
fallback_image="boot/initramfs-linuxtest-fallback.img"
fallback_options="-S autodetect"
```

然后运行命令`sudo mkinitcpio -p linuxtest`编译就完成了

C

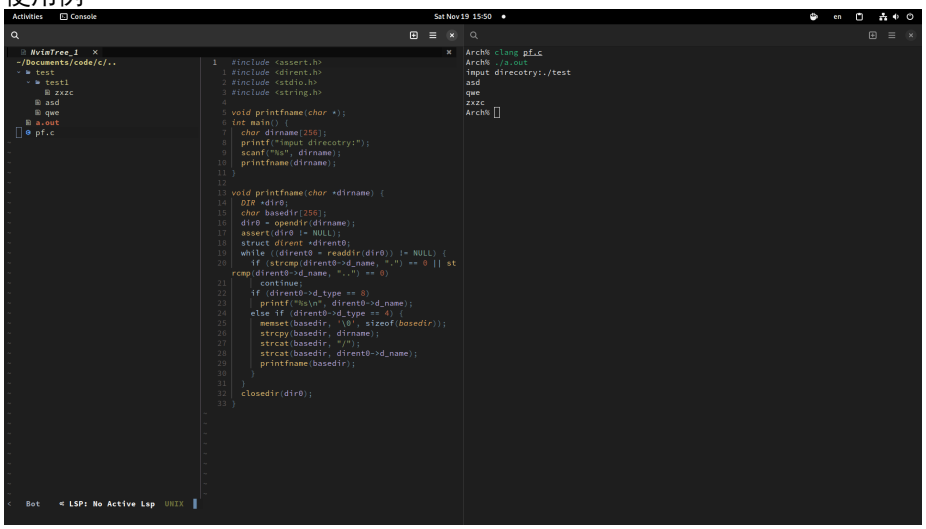
输出路径下文件

代码如下

```
#include <assert.h>
#include <dirent.h>
#include <stdio.h>
#include <string.h>

void printfname(char *);
int main() {
    char dirname[256];
    printf("imput direcotry:");
    scanf("%s", dirname);
    printfname(dirname);
}

void printfname(char *dirname) {
    DIR *dir0;
    char basedir[256];
    dir0 = opendir(dirname);
    assert(dir0 != NULL);
    struct dirent *dirent0;
```

```
    while ((dirent0 = readdir(dir0)) != NULL) {
        if (strcmp(dirent0->d_name, ".") == 0 || strcmp(dirent0->d_name, "..") == 0)
            continue;
        if (dirent0->d_type == 8)
            printf("%s\\n", dirent0->d_name);
        else if (dirent0->d_type == 4) {
            memset(basedir, '\0', sizeof(basedir));
            strcpy(basedir, dirname);
            strcat(basedir, "/");
            strcat(basedir, dirent0->d_name);
            printfname(basedir);
        }
    }
    closedir(dir0);
}
```

使用例



注

这个pdf是用pandoc生成的，我找了很久也没找到一个能和nvim配合比较好的pdf生成器
我对pandoc并不是很满意，首先和离开了微软雅黑和苹方的大多数linux软件一样，
它对中文的支持并不是很好
为了能获得一个较为统一的显示效果，代码里的注释全都被我重写成了蹩脚的工地英语
再者我感觉pandoc的bug很多，比如

```
Arch% pandoc --pdf-engine=xelatex -V mainfont='WenQuanYi Zen Hei' linux.md -o coiquan.pdf
Error producing PDF.
! Undefined control sequence.
l.265 d\_type == 8) printf("\%s\n
```

最后是把md源文档改成这个样才过的

```
    if (dirent0->d_type == 8)
        printf("%s\\n", dirent0->d_name);
    else if (dirent0->d_type == 4) {
```