Build your own Windows+Linux+NAS development environment

原创 jiangwei0512 Posted on 2019-01-27 13:37:46 Read 1.1w Collection 23 Likes 2

Category Column: Miscellaneous Article Tags: NAS

摘要 This article introduces a solution for building a cross-platform development environment, aiming to use Windows as the main development environment, Linux as the auxiliary compilation and execution environment, and NAS for code storage and interaction. It involves the configuration of Windows 10, Ubuntu 18.04 and NAS, including key steps such as SSH connection, compiler installation, and NAS access settings.

The summary is generated in C Know , supported by DeepSeek-R1 full version, go to experience>

background

Some development requires the use of both Windows and Linux environments. Switching between multiple machines is troublesome, so here we try to build a development environment with Windows as the main development environment, Linux as the auxiliary compilation and execution environment, and NAS as code storage and interaction.

Prepare

A Windows host (Windows 10), a Linux host (Ubuntu18.04), and a NAS.

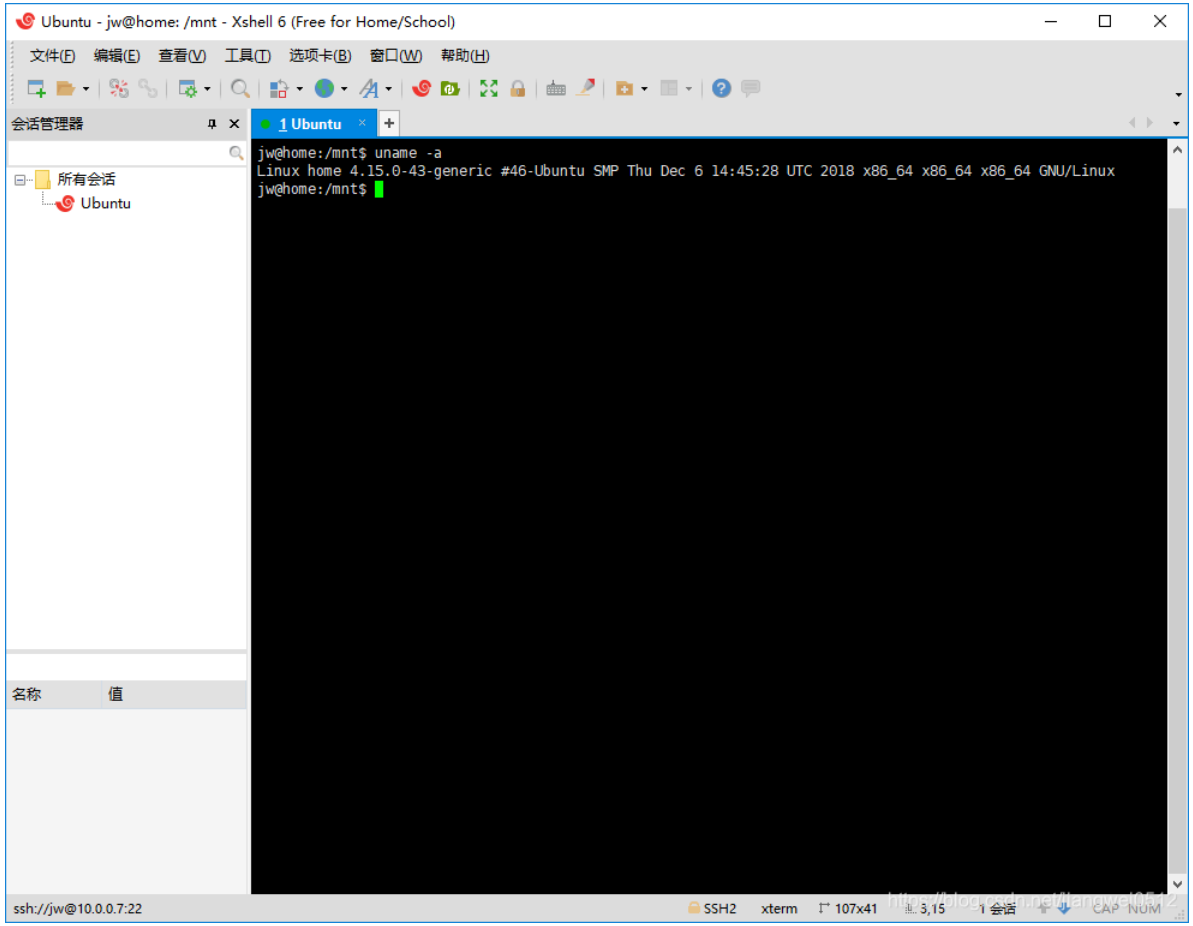
Environment Construction

It needs to be explained separately for different ends.

Windows 10

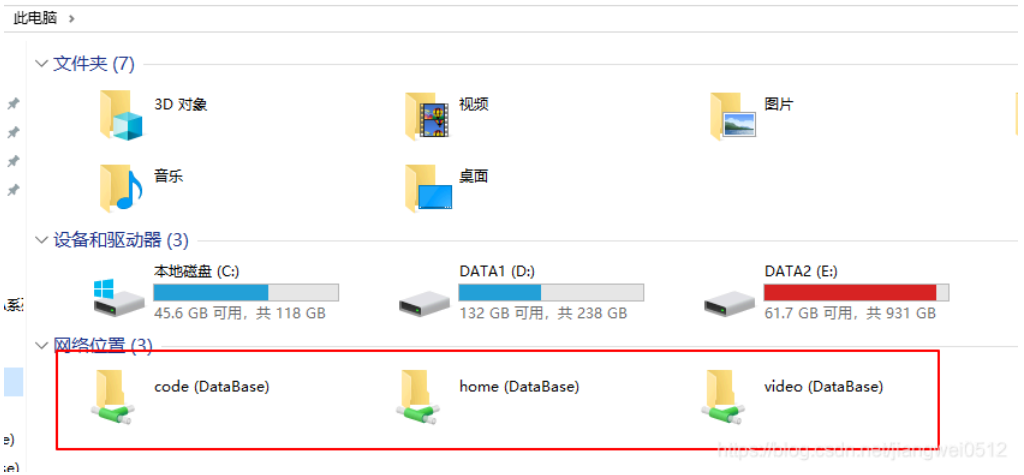
1. Code editor and compiler. Since I am familiar with development under Windows, I can use editors and compilers smoothly. The specific editor and compiler to use depends on the individual. Here I mainly use VS Code and VS Studio 2013.

2. SSH connection tool, used to connect to the Linux server. Here we choose XShell, which has a free version. The following is the interface for connecting to Ubuntu:

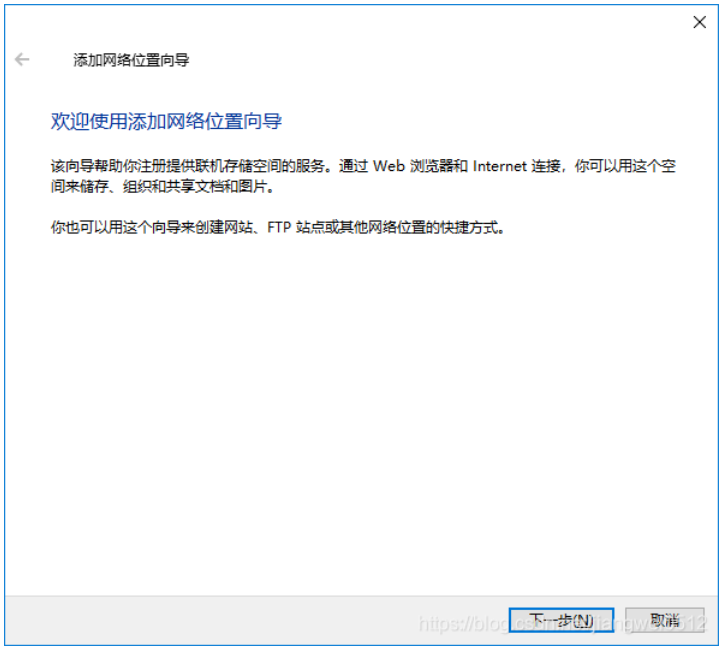


If you need to compile under Linux, do it through this interface.

3. Create a NAS access directory (SMB needs to be configured on the NAS side, see below), right-click on the following interface and select Create Network Location:



The red box is already created. You can create a network location step by step according to the instructions in the figure below:



Linux

1. First, in order to access the Linux side through SSH under Windows, you need to install the SSH server on the Linux side. By default, Ubuntu 18.04 does not have an SSH server installed, so you need to install it manually. The installation command is as follows:

bash AI generated projects 登录复制

```
sudo apt install openssh-server
```

After the installation is complete, you need to open:

bash AI generated projects 登录复制

```
sudo /etc/init.d/ssh start
```

2. The next step is to install compilers such as GCC and other tools according to actual usage.

3. In order to access NFS under Linux, you also need to install NFS tools:

bash AI generated projects 登录复制

```
sudo apt install nfs-common nfs-kernel-server
```

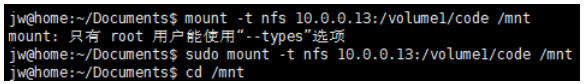
After that, you can mount NFS through commands (NFS needs to be configured on the NAS side, see below):

bash AI generated projects 登录复制

```
mount -t nfs 10.0.0.13:/volume1/code /mnt
```

After that, you can access the data corresponding to the NAS (the IP and directory should be modified according to the actual situation).

There is still a problem. The above mount can only be done under root privileges. If you mount under user privileges, you will not be able to enter the mnt directory afterwards:



NAS

The NAS side is mainly used as a data storage transfer station for Windows and Linux sides, and it needs to ensure that both Windows and Linux can access it normally.

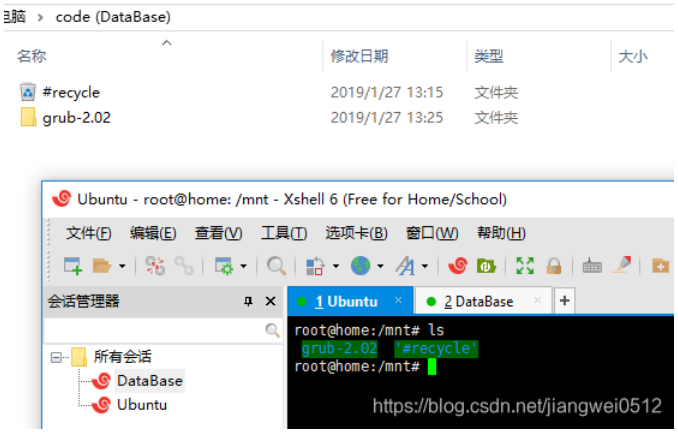
The NAS used here is Synology's. The operations of NAS of different brands may be different. Here we only take the Synology DS218Play model device as an example.

1. Open SMB:

Then you can access it through \\DataBase in Windows (DataBase is the name of the NAS here and needs to be modified according to actual conditions).

2. Enable NFS:

Below is a diagram of the access code under Windows and Linux after the configuration is completed:



postscript

The environment built this time can be guaranteed to be used, but the cost, security, performance and other issues have not been considered yet.