

FuSa Compiler安装激活手册

目录

安装要求	2
软件下载	2
单机版	3
一、 下载 FuSa Compiler license 文件	3
1.1 注册 ARM 帐户	3
1.2 登录 ARM 帐户，下载 License 文件	5
二、 软件安装	8
2.1 Windows 版安装	8
2.2 Linux 版安装	8
三、 激活软件	8
3.1 Windows 版	8
3.2 Linux 版	10
网络版	11
一、 下载 FuSa Compiler license 文件	11
1.1 注册 ARM 帐户	11
1.2 登录 ARM 帐户，下载 License 文件	13
二、 license 合并	17
三、 软件安装	17
3.1 Windows 版安装	17
3.2 Linux 版安装	18
四、 服务器端管理工具安装	18
Windows 版	<i>Error! Bookmark not defined.</i>
五、 激活软件	21
5.1 Windows 版	21
5.2 Linux 版	22

安装要求

操作系统			硬件要求
Windows 10	红帽	Ubuntu	双核2GHZ及以上
	企业版 Linux 7工作站	桌面版18.04 LTS	4GB RAM及以上
		桌面版 20.04 LTS	4GB以上可用 硬盘空间

软件下载

Arm compiler 软件下载

<https://developer.arm.com/downloads/-/arm-compiler-for-functional-safety>

FlexNet 软件下载

<https://developer.arm.com/downloads/-/downloads>

单机版

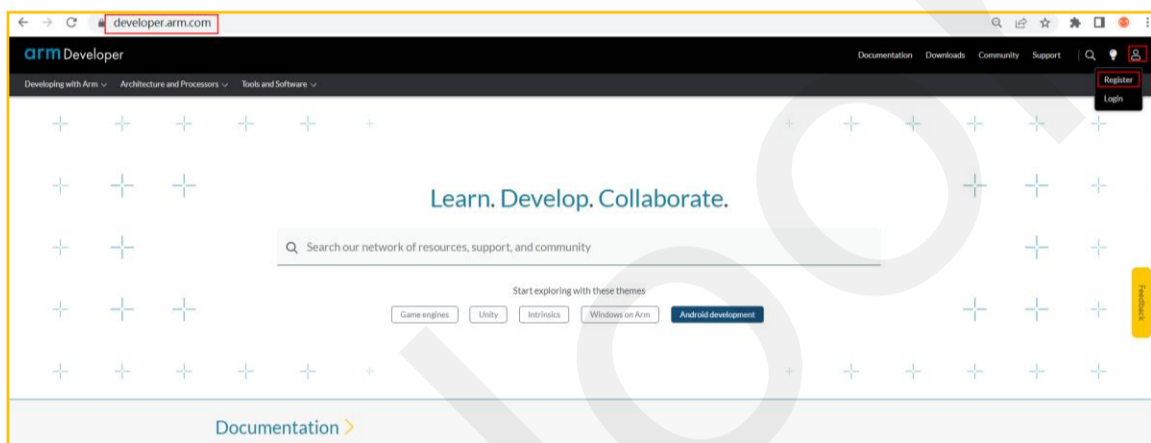
一、 下载 FuSa Compiler license 文件

1.1 注册 ARM 帐户

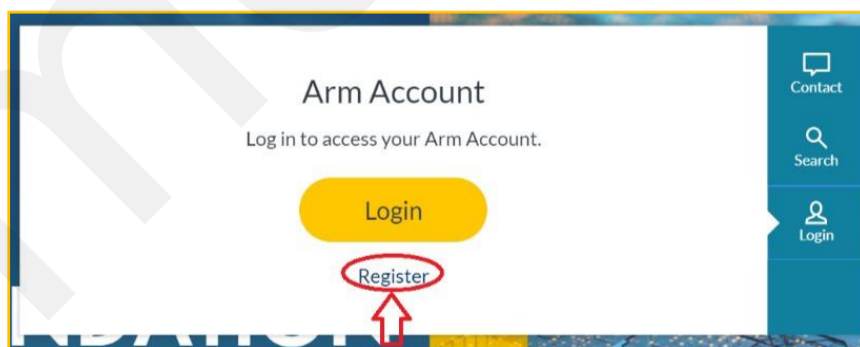
如果您已经有 ARM 帐户了，请跳过”注册账号”这一节，直接看第二节”登录 ARM 帐户，下载 License 文件”

如果打开 ARM 官网太慢，建议用笔记本电脑连接手机热点来访问 ARM 官网

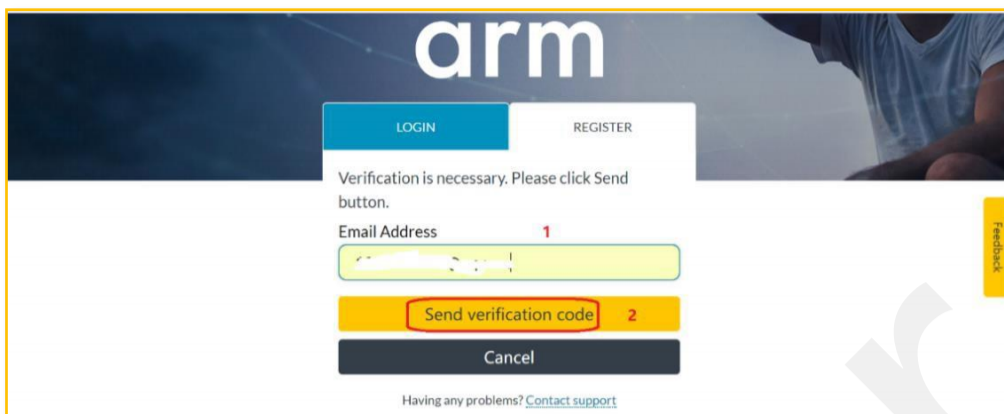
- 1) 打开浏览器在地址栏中输入 <https://developer.arm.com/> 进入 ARM 官网



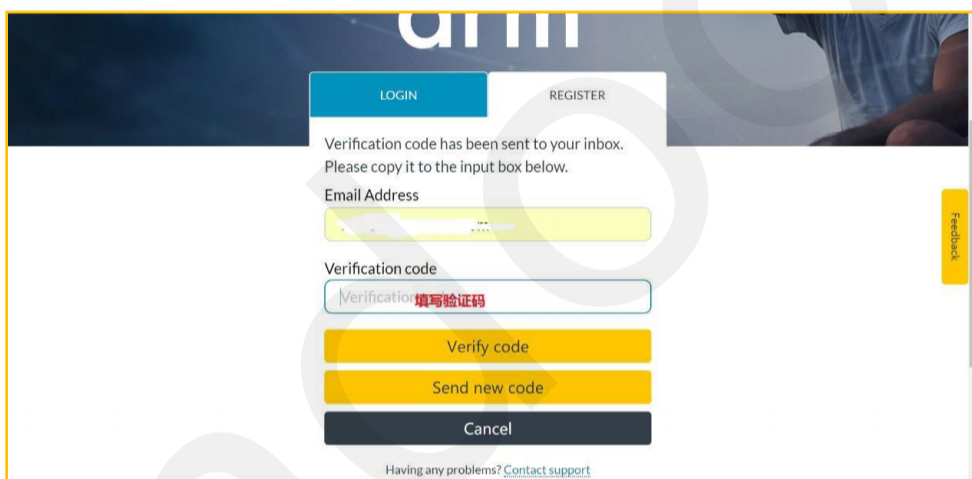
- 2) 点击右上角  人物图标, 选择 Register 进入注册页面



- 3) 在 Email address 栏处输入需要注册的邮箱地址，然后点击 Send verification code 按钮，向填写的邮箱发送一份带有验证码的邮件。

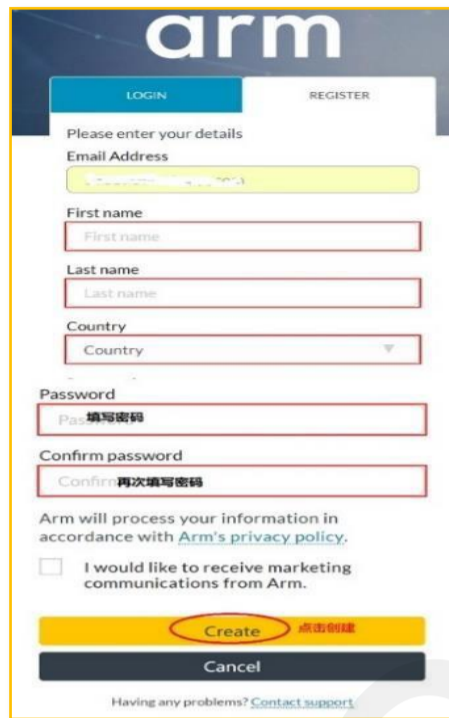


- 4) 收取邮件并将验证码填写到 Verification code 栏处，然后点击 Verify code 按钮。



注意：如果您没有收到验证码可以点击 Send new code 按钮重新向刚才填写的邮箱中发送验证码;如果还是没有收到邮件，则有可能是邮箱过滤了一些陌生邮件，请到垃圾邮件中查看。但垃圾邮件中还没找到时，则可能是邮箱被设置了拦截功能请联系 IT 帮忙处理。

5) 填写相关信息，然后点击 Create 按钮创建帐户



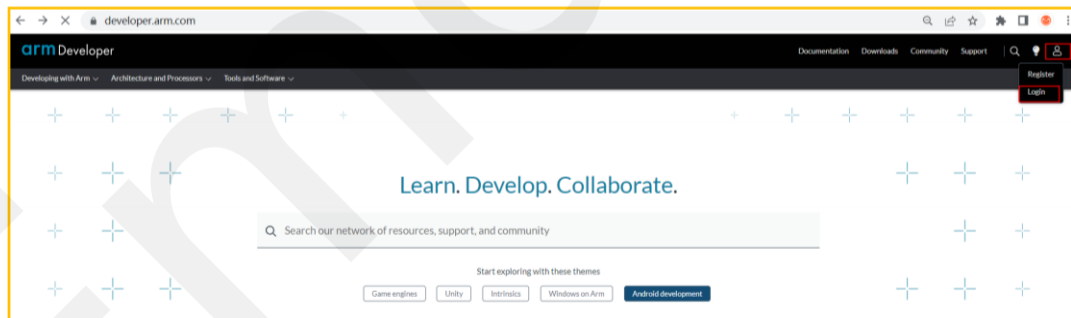
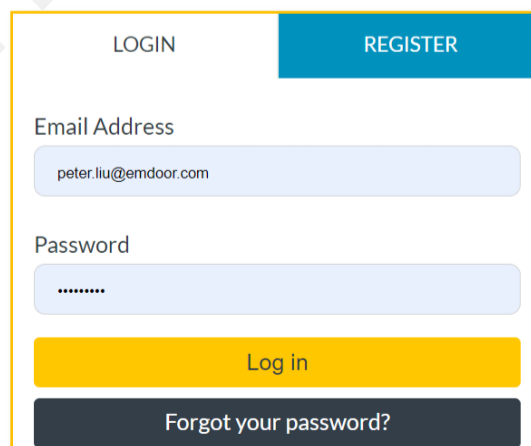
The image shows the ARM Developer registration form. It has a header with the ARM logo and tabs for LOGIN and REGISTER. The form asks for the following details:

- Email Address (with a placeholder email address)
- First name
- Last name
- Country (dropdown menu)
- Password (with a placeholder '填写密码')
- Confirm password (with a placeholder '再次填写密码')

Below the form, there is a checkbox for 'I would like to receive marketing communications from Arm.' and a 'Create' button (labeled '点击创建'). A 'Cancel' button is also present. At the bottom, there is a link for 'Contact support'.

1.2 登录 ARM 帐户，下载 License 文件

1) 在浏览器的地址栏中输入<https://developer.arm.com/> 进入 ARM 官网，右上角人物图标点击Login in 按钮。

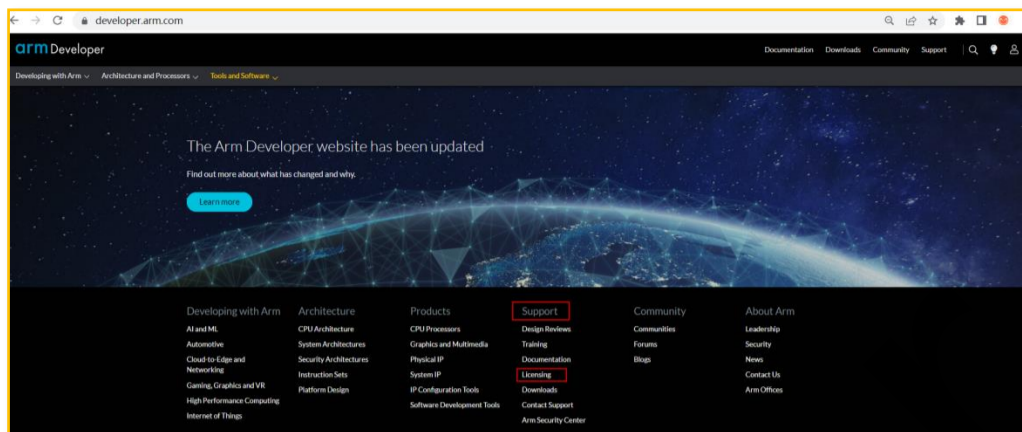



The image shows the ARM Developer login form. It has a header with tabs for LOGIN and REGISTER. The form asks for the following details:

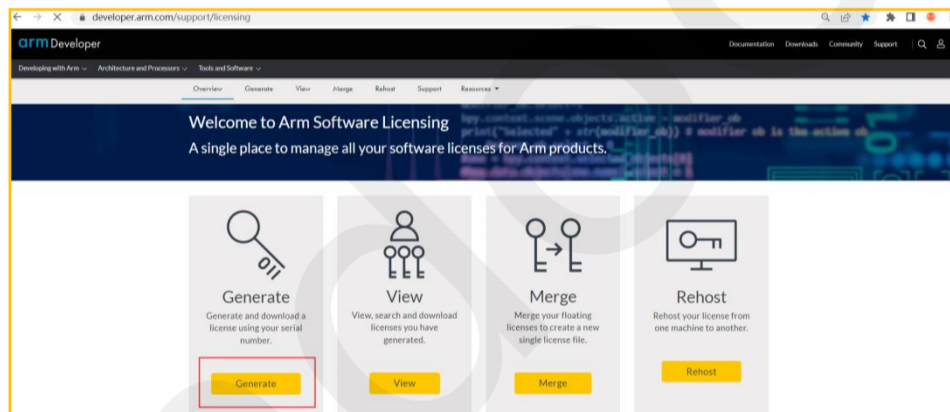
- Email Address (with the placeholder email address 'peter.liu@emdoor.com')
- Password (with a placeholder '*****')

Below the form, there is a 'Log in' button and a 'Forgot your password?' link.

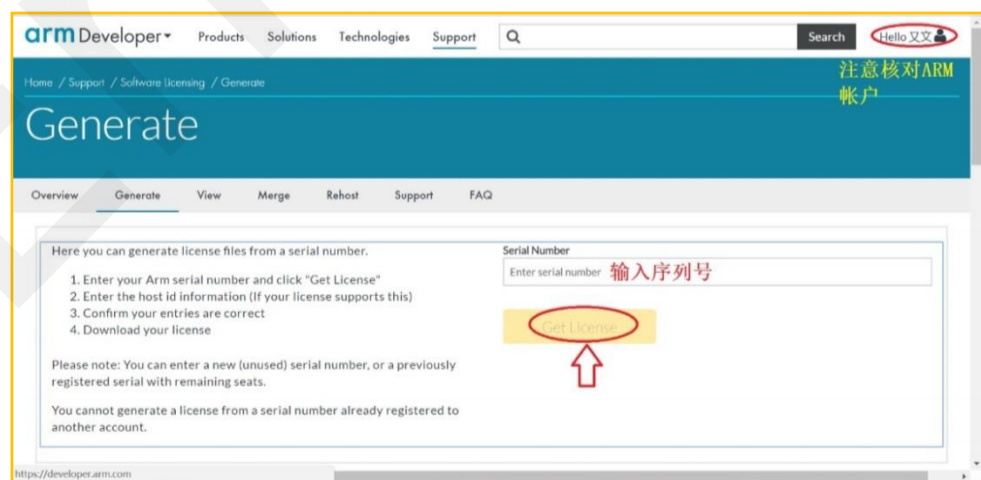
2) 向下拖动垂直滑动条，找到 Support-->License按钮。



3) 点击 licensing 按钮，找到 Generate 按钮。



4) 输入序列号，然后点击 Get License 按钮生成



- 5) 选择 HostID Type 处选择类型为 ETHERNET，在 HostID 处填写服务器的 MAC 地址，然后点击 Generate 按钮进入确认页面。核对下 MAC 地址确认无误后点击 Confirm 按钮。

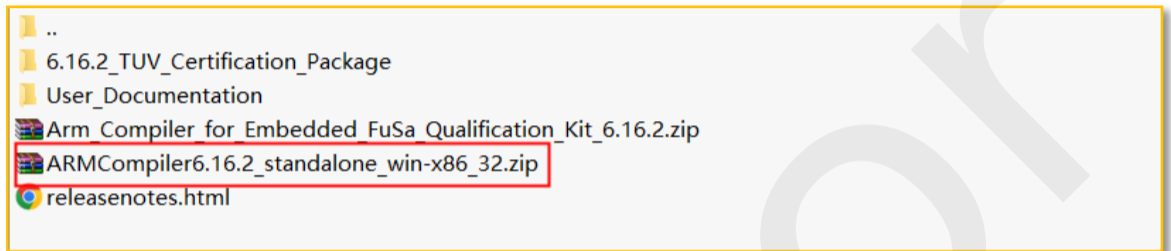
- 6) 点击 Download License 按钮下载 license 文件，如果没有出现下载窗口请设置浏览器允许弹窗。

- 7) 在打开的页面中点击 Save 按钮，下载 License 文件。

二、 软件安装

2.1 Windows 版安装

解压 ARM Compiler 文件到自定义目录，双击运行 `setup.exe` 进行 ARM Compiler 程序的安装。



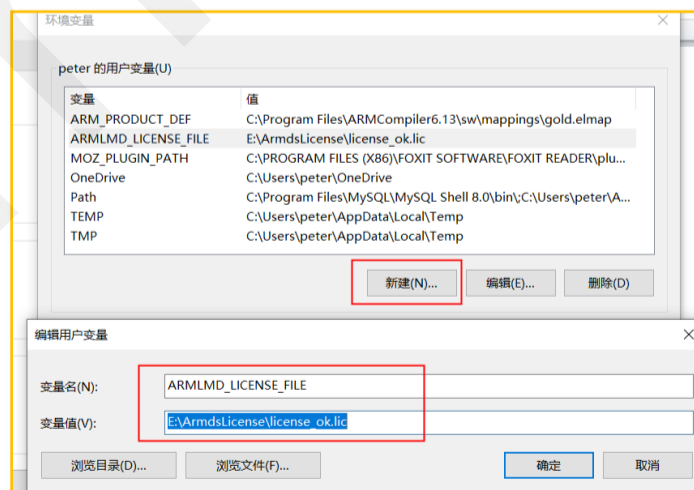
2.2 Linux 版安装

解压并切换到 Arm compiler 解压的目录中，然后在命令行终端中输入 `./install_x86_64.sh` 就行。

三、 激活软件

3.1 Windows 版

- 1) 设置 License 环境变量 `ARMLMD_LICENSE_FILE`，
`ARMLMD_LICENSE_FILE=license` 文件路径，如：



2) 验证:

在 cmd 窗口中切换到arm 编译器路径, 输入编译器名称并回车。比如: `armclang --help`

```
D:\Keil_v537\ARM\ArmCompilerforEmbeddedFuSa6.16.2\bin>armclang --help
Product:
Component: Arm Compiler for Embedded FuSa 6.16.2
Tool: armclang [5e00ae00]

Target: unspecified-arm-none-unspecified

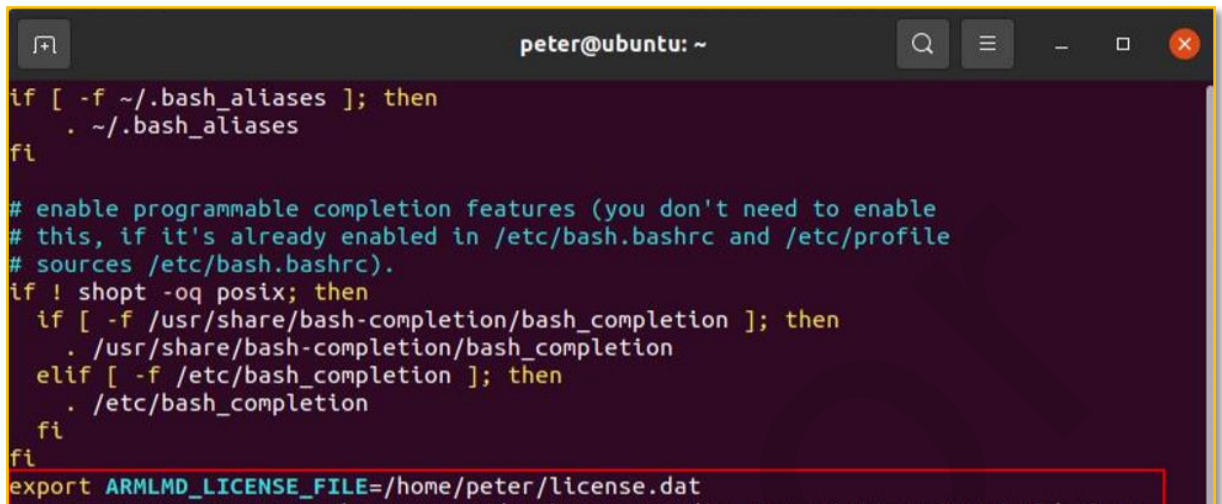
Usage: armclang --target=<triple> [options] file1 file2 ... fileN

Mandatory options:
--target          Generate code for the specified target triple. Supported values are:
                  "arm-arm-none-eabi" - generate code for AArch32 state
                  "aarch64-arm-none-eabi" - generate code for AArch64 state

Compiler and integrated assembler options:
-march            Targets an architecture, generating generic code that runs on any
                  processor of that architecture.
-mcpu            Targets a specific processor, generating optimized code for that
                  specific processor.
-mfpu            Targets a specific floating-point unit.
-march=list       Show a list of available architectures, CPUs and FPUs.
-mcpu=list
-mfpu=list
-mbig-endian      Generate code for a big-endian processor.
-mlittle-endian   Generate code for a little-endian processor.
-mimplicit-it     Specifies the behavior of the integrated assembler if there are
                  conditional instructions outside IT blocks.
```

3.2 Linux 版

- 1) 在用户目录下的“.bashrc”文件中添加“ARMLMD_LICENSE_FILE”环境变量的相关内容，如下图所示：

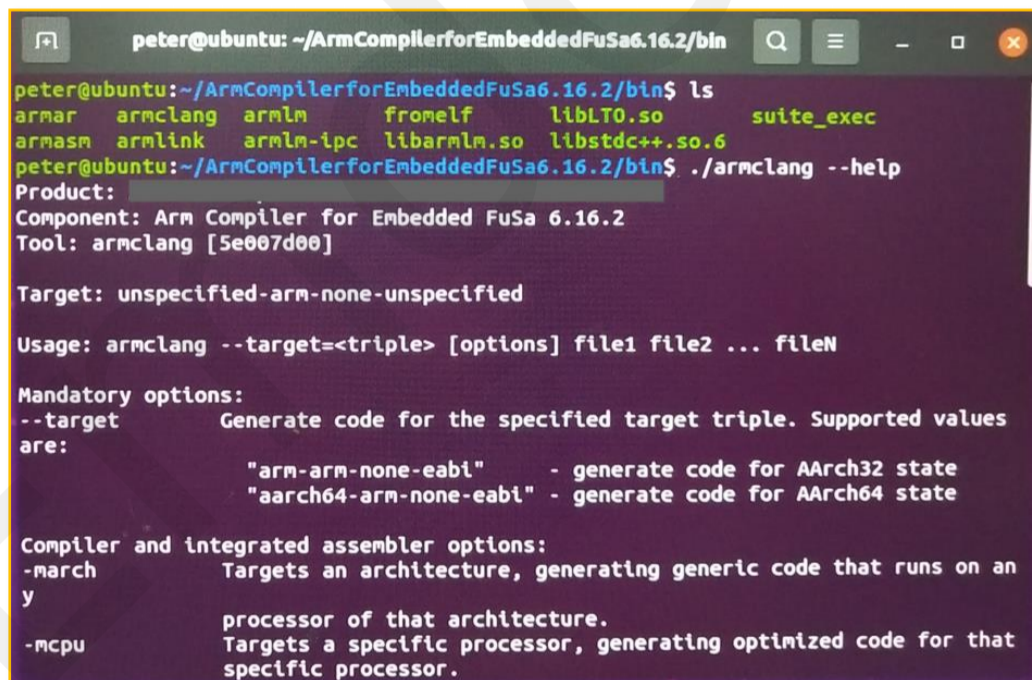


```
peter@ubuntu: ~  
if [ -f ~/.bash_aliases ]; then  
    . ~/.bash_aliases  
fi  
  
# enable programmable completion features (you don't need to enable  
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile  
# sources /etc/bash.bashrc).  
if ! shopt -oq posix; then  
    if [ -f /usr/share/bash-completion/bash_completion ]; then  
        . /usr/share/bash-completion/bash_completion  
    elif [ -f /etc/bash_completion ]; then  
        . /etc/bash_completion  
    fi  
fi  
export ARMLMD_LICENSE_FILE=/home/peter/license.dat
```

ARMLMD_LICENSE_FILE=license 文件的绝对路径

- 2) 验证:

在 cmd 窗口中切换到arm 编译器路径，输入编译器名称并回车。比如：armclang --help



```
peter@ubuntu: ~/ArmCompilerforEmbeddedFuSa6.16.2/bin  
peter@ubuntu:~/ArmCompilerforEmbeddedFuSa6.16.2/bin$ ls  
armar  armclang  armlm  fromelf  libLTO.so  suite_exec  
armasm armlink  armlm-ipc libarmlm.so libstdc++.so.6  
peter@ubuntu:~/ArmCompilerforEmbeddedFuSa6.16.2/bin$ ./armclang --help  
Product:  
Component: Arm Compiler for Embedded FuSa 6.16.2  
Tool: armclang [5e007d00]  
  
Target: unspecified-arm-none-unspecified  
  
Usage: armclang --target=<triple> [options] file1 file2 ... fileN  
  
Mandatory options:  
--target Generate code for the specified target triple. Supported values  
are:  
          "arm-arm-none-eabi" - generate code for AArch32 state  
          "aarch64-arm-none-eabi" - generate code for AArch64 state  
  
Compiler and integrated assembler options:  
-march Targets an architecture, generating generic code that runs on an  
y  
processor of that architecture.  
-mcpu Targets a specific processor, generating optimized code for that  
specific processor.
```

网络版

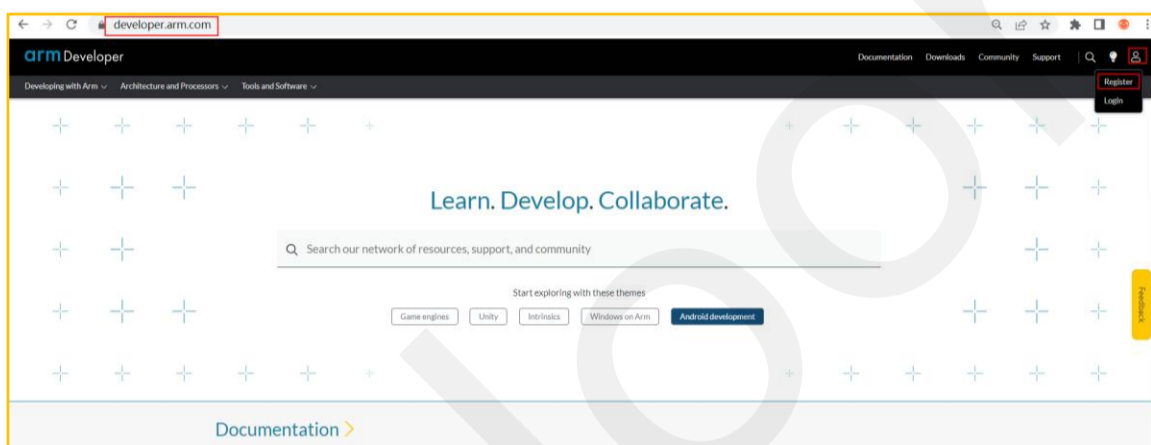
一、 下载 FuSa Compiler license 文件

1.1 注册 ARM 帐户

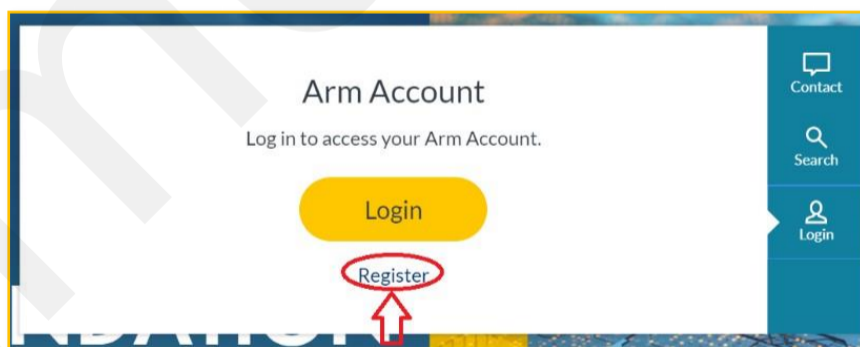
如果您已经有 ARM 帐户了，请跳过”注册账号”这一节，直接看第二节”登录 ARM 帐户，下载 License 文件”

如果打开 ARM 官网太慢，建议用笔记本电脑连接手机热点来访问 ARM 官网

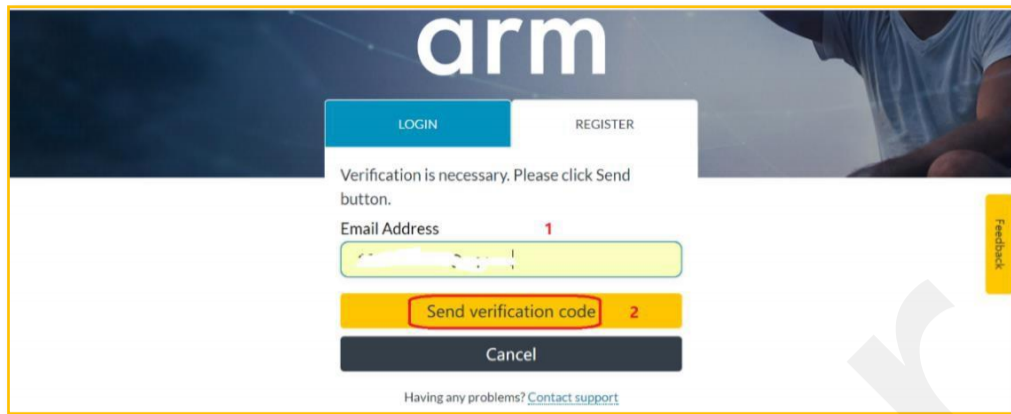
1) 打开浏览器在地址栏中输入 <https://developer.arm.com/> 进入 ARM 官网



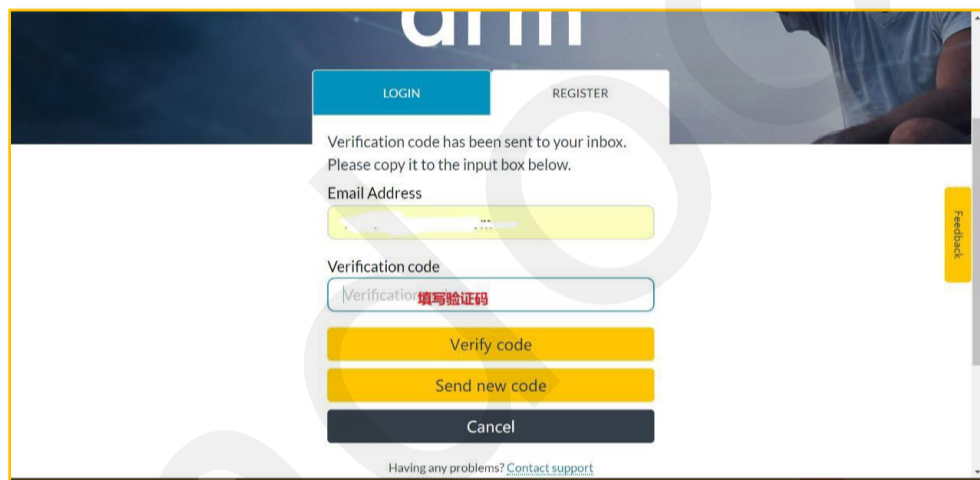
2) 点击右上角  人物图标, 选择 Register 进入注册页面



- 3) 在 Email address 栏处输入需要注册的邮箱地址，然后点击 Send verification code 按钮，向填写的邮箱发送一份带有验证码的邮件。



- 4) 收取邮件并将验证码填写到 Verification code 栏处，然后点击 Verify code 按钮。



注意：如果您没有收到验证码可以点击 Send new code 按钮重新向刚才填写的邮箱中发送验证码;如果还是没有收到邮件，则有可以是邮箱过滤了一些陌生邮件，请到垃圾邮件中查看。但垃圾邮件中还没找到时，则可能是邮箱被设置了拦截功能请联系 IT 帮忙处理。

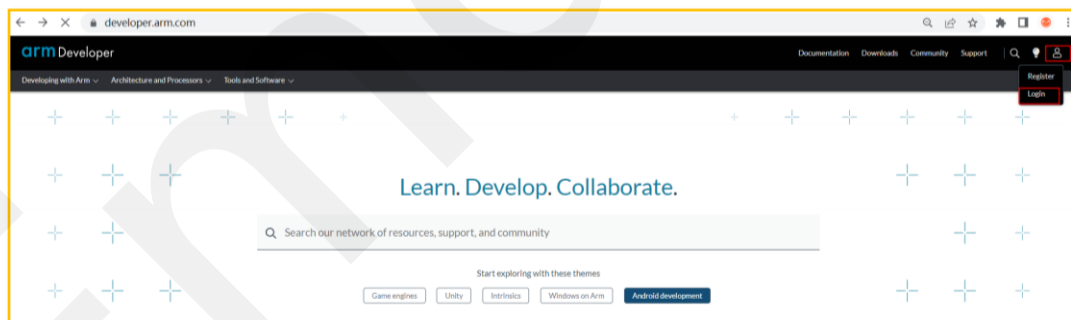
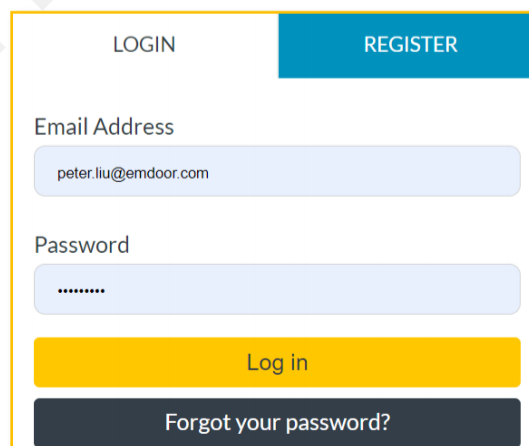
5) 填写相关信息，然后点击 Create 按钮创建帐户



The image shows the ARM Developer registration form. It has a header with 'arm' logo and 'LOGIN' and 'REGISTER' tabs. The 'REGISTER' tab is active. The form asks for 'Email Address', 'First name', 'Last name', 'Country' (a dropdown menu), 'Password', and 'Confirm password'. There is a checkbox for 'I would like to receive marketing communications from Arm.' and a 'Create' button (labeled '点击创建' in Chinese). A 'Cancel' button is also present. At the bottom, there is a link for 'Contact support'.

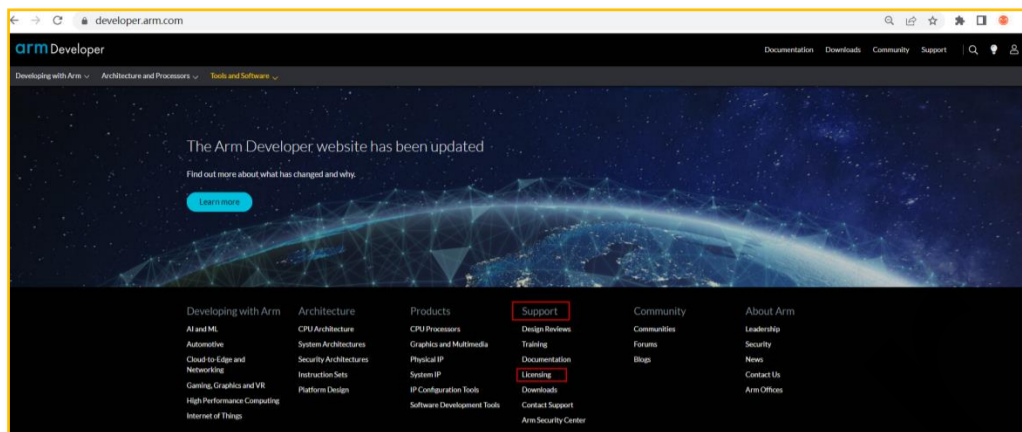
1.2 登录 ARM 帐户，下载 License 文件

1) 在浏览器的地址栏中输入<https://developer.arm.com/> 进入 ARM 官网，右上角  人物图标点击 **Login in** 按钮。

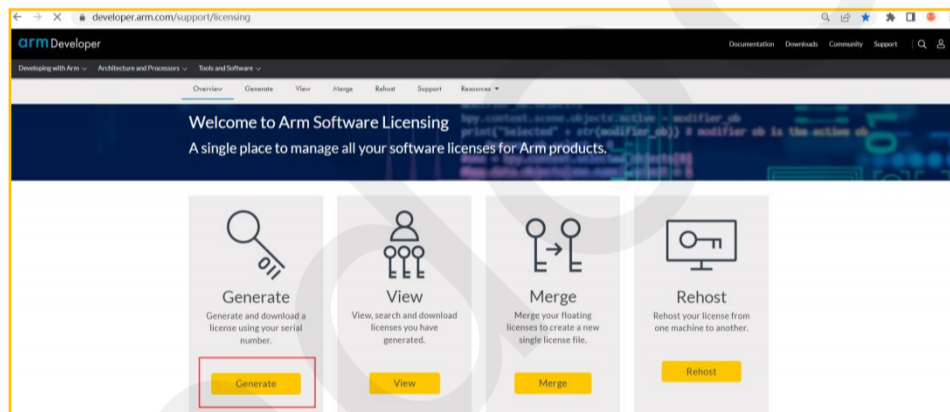



The image shows the ARM Developer login form. It has a header with 'LOGIN' and 'REGISTER' tabs. The 'LOGIN' tab is active. The form asks for 'Email Address' (with the example 'peter.liu@emdoor.com') and 'Password'. There is a 'Log in' button and a 'Forgot your password?' link.

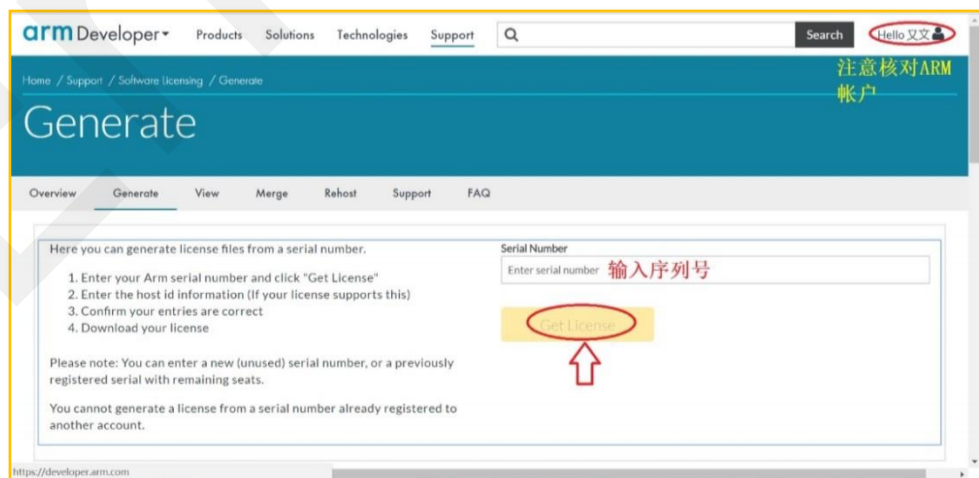
2) 向下拖动垂直滑动条，找到 Support-->License按钮。



3) 点击 licensing 按钮，找到 Generate 按钮。



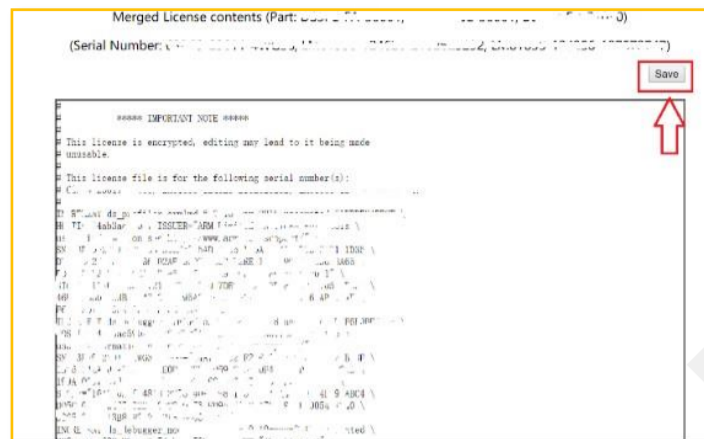
4) 输入序列号，然后点击 Get License 按钮生成



- 5) 选择 HostID Type 处选择类型为 ETHERNET，在 HostID 处填写服务器的 MAC 地址，然后点击 Generate 按钮进入确认页面。核对下 MAC 地址确认无误后点击 Confirm 按钮。

- 6) 点击 Download License 按钮下载 license 文件，如果没有出现下载窗口请设置浏览器允许弹窗。

- 7) 在打开的页面中点击 Save 按钮，下载 License 文件。



- 8) License文件的修改，把“this_host”修改为服务器的IP，在MAC地址后，添加端口号：8224。
注：MAC与端口号之间需要有空格。

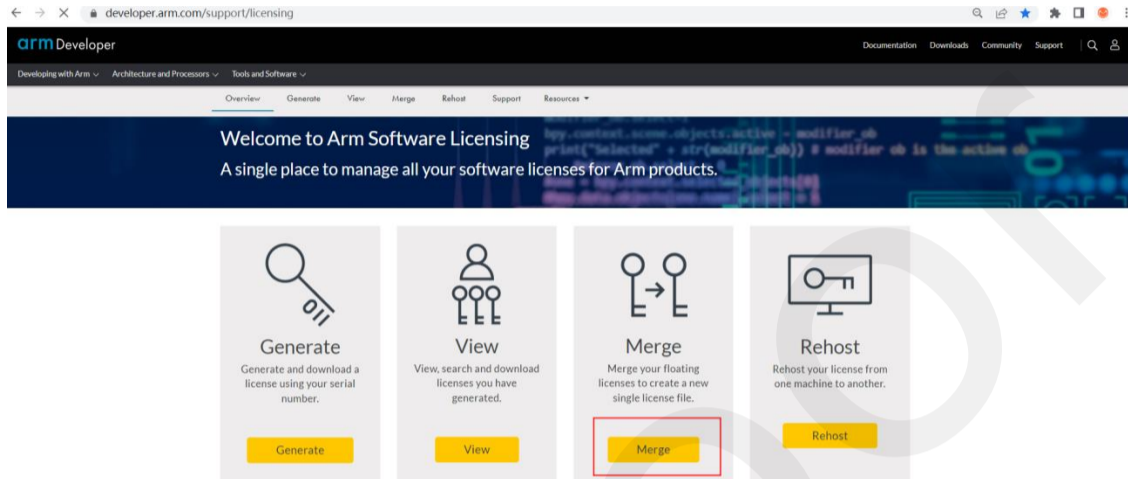
```
#
# For example if you have obtained this license for RVDS 4.1 and you
# have an existing ADS license server, then you will need to upgrade
# to the latest armlmd and lmgrd daemons which can be downloaded from
# https://developer.arm.com/flexnet-binaries
#
SERVER this_host XXXXXXXXXX
VENDOR armlmd
USE_SERVER
```

```
#
# For example if you have obtained this license for RVDS 4.1 and you
# have an existing ADS license server, then you will need to upgrade
# to the latest armlmd and lmgrd daemons which can be downloaded from
# https://developer.arm.com/flexnet-binaries
#
SERVER 172.30.2.130 XXXXXXXXXX 8224
VENDOR armlmd
USE_SERVER
```

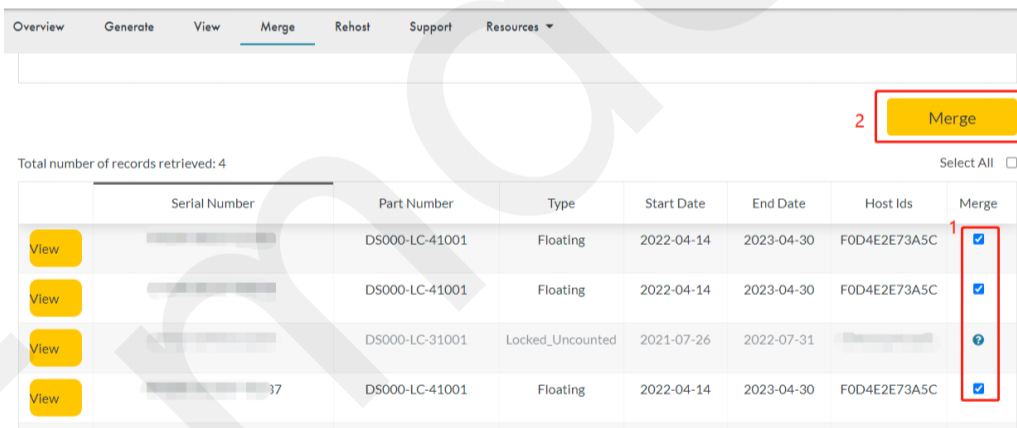
二、 license 合并

注意：如果您购买的 arm 软件有多个序列号且都需要部署在同一台服务器上就需要合并这些 license 文件。如果不是这种情况请忽略 " **License 合并** " 这一节

- 1) 在浏览器的地址栏中输入<https://developer.arm.com/> 进入 ARM 官网, 登录 arm 账号, 进入 "<https://developer.arm.com/support/licensing>" 页面, 点击"Merge"



- 2) 勾选需要合并的 license, 注意: 只有 MAC 地址一样的才能合并。点击 Merge 按钮按照向导完成合并



三、 软件安装

3.1 Windows 版安装

解压 ARM Compiler 文件到自定义目录, 双击运行 `setup.exe` 进行 ARM Compiler 程序的安装。

3.2 Linux 版安装

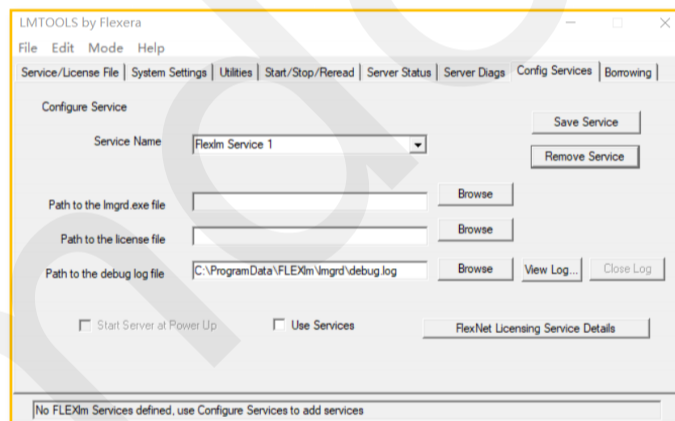
解压并切换到 Arm compiler 解压的目录中，然后在命令行终端中输入./ install_x86_64.sh就行。

四、 服务器端管理工具安装

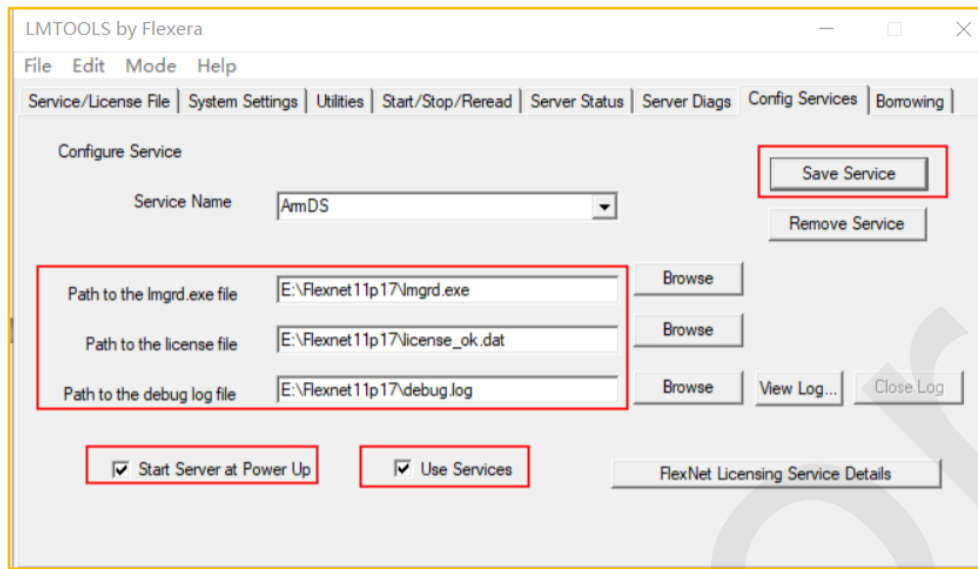
1) 我们以 X86-64 位的Flexnet工具软件为例，下载Flexnet工具后，需要解压，解压后打开文件夹。如下图：



2) 双击 lmttools.exe，出现以下对话框，点击 Config Services。



3) 点击 Config Services, 开始配置服务。



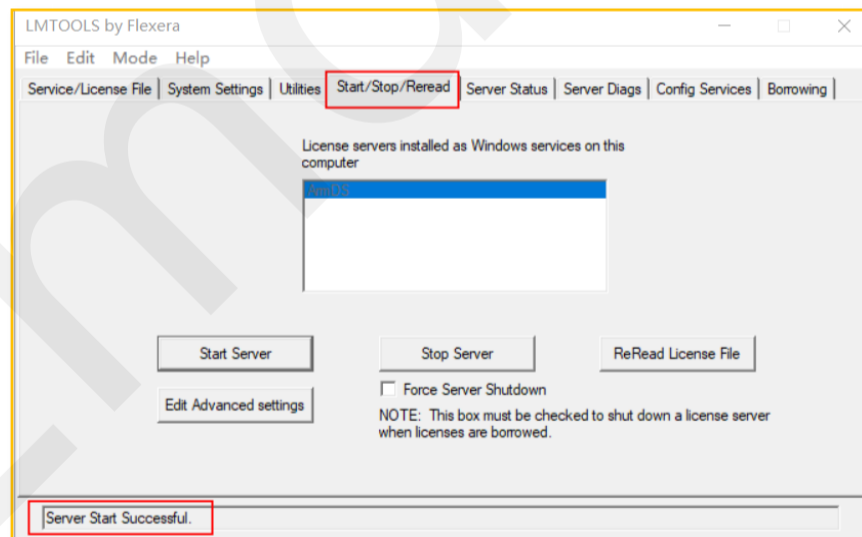
注：选填各个路径，所有的文件最好都在同一个目录下：

lmgrd.exe 在下载的软件的文件夹里

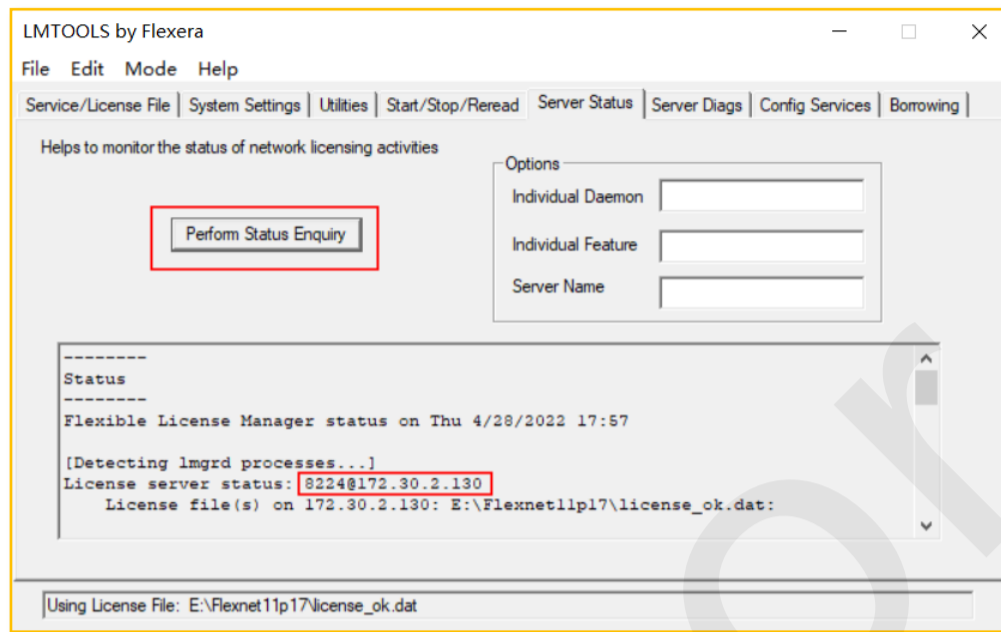
license.dat从官方网站下载

debug log file 需要新建。

4) 以上步骤完成后，点击 save service 保存操作。点击 Start/Stop/Reread栏，点击 Start Server，启动服务（**Server Starts Successfully**）。



5) 用户可点击 **Server Status** 查看服务状态。

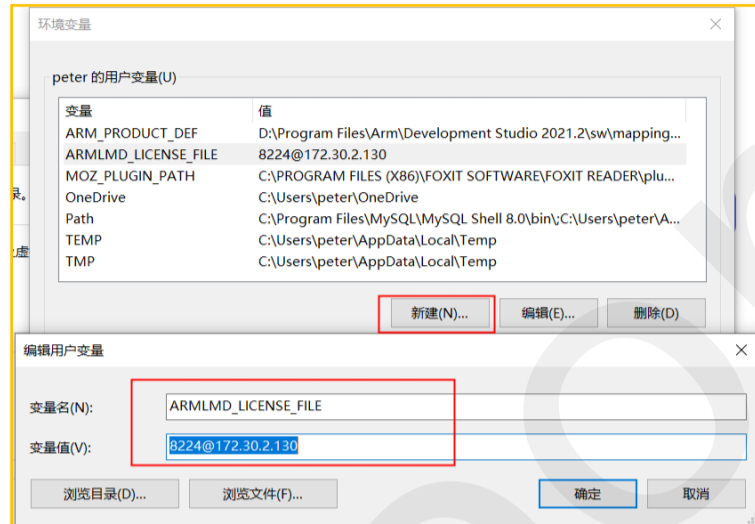


点击 **Perform Status Enquiry** 下方会出现当前license的状态。到此，服务器端安装完成。

五、 激活软件

5.1 Windows 版

1) 设置 License 环境变量 ARMLMD_LICENSE_FILE
ARMLMD_LICENSE_FILE=license 文件路径, 如:



2) 验证:

在 cmd 窗口中切换到arm 编译器路径, 输入编译器名称并回车。比如: `armclang --help`

```
D:\Keil_y537\ARM\ArmCompilerforEmbeddedFuSa6.16.2\bin>armclang --help
Product:
Component: Arm Compiler for Embedded FuSa 6.16.2
Tool: armclang [5e00ae00]

Target: unspecified-arm-none-unspecified

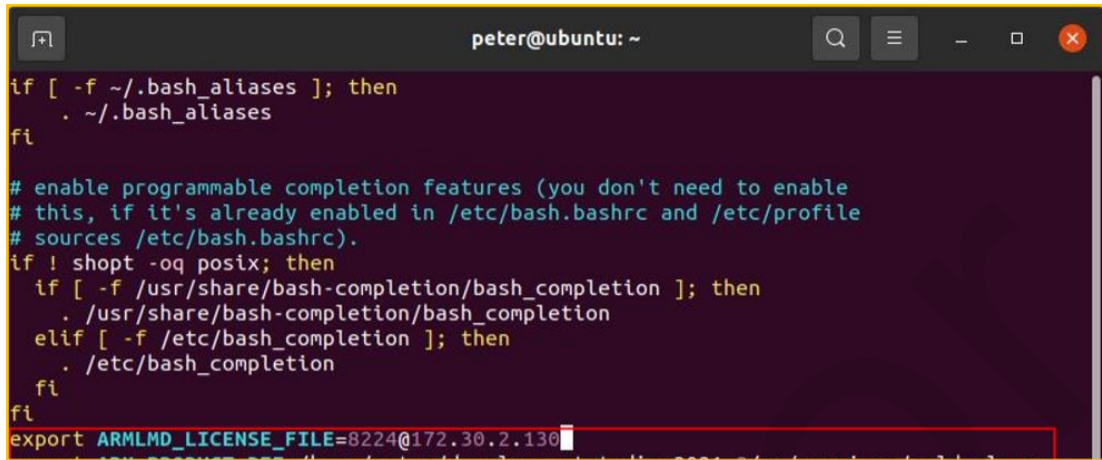
Usage: armclang --target=<triple> [options] file1 file2 ... fileN

Mandatory options:
--target      Generate code for the specified target triple. Supported values are:
               "arm-arm-none-eabi" - generate code for AArch32 state
               "aarch64-arm-none-eabi" - generate code for AArch64 state

Compiler and integrated assembler options:
-march        Targets an architecture, generating generic code that runs on any
               processor of that architecture.
-mcpu         Targets a specific processor, generating optimized code for that
               specific processor.
-mfpu         Targets a specific floating-point unit.
-march=list   Show a list of available architectures, CPUs and FPU's.
-mcpu=list    Generate code for a big-endian processor.
-mbig-endian  Generate code for a little-endian processor.
-mimplicit-it Specifies the behavior of the integrated assembler if there are
               conditional instructions outside IT blocks.
```

5.2 Linux 版

- 1) 在用户目录下的 “.bashrc” 文件中添加“ARMLMD_LICENSE_FILE”环境变量的相关内容，如下图所示：

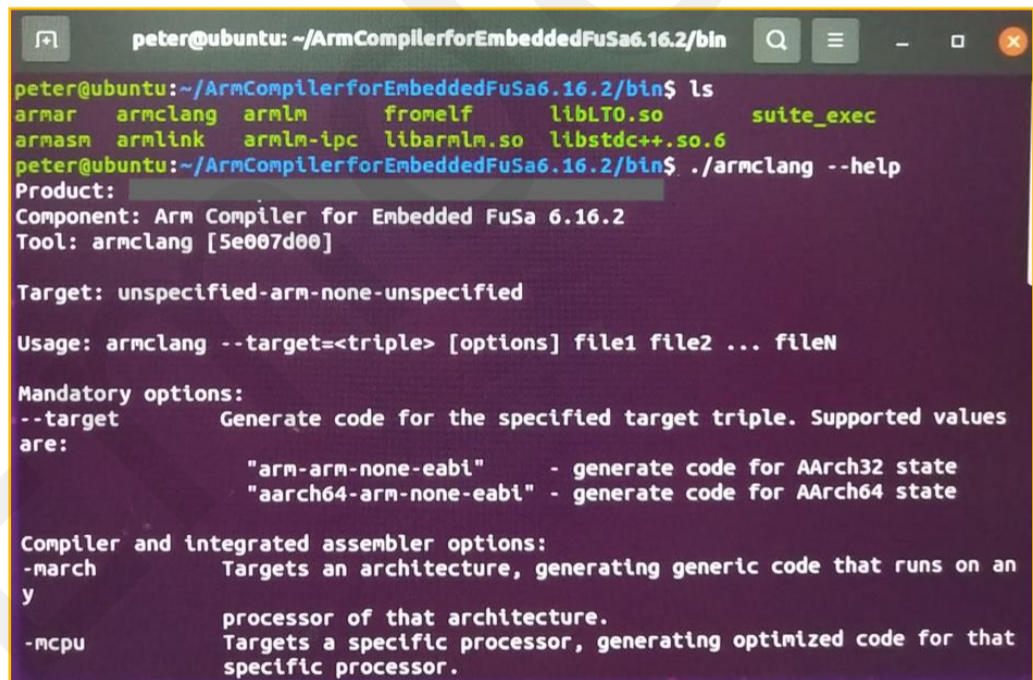


```
peter@ubuntu: ~  
if [ -f ~/.bash_aliases ]; then  
    . ~/.bash_aliases  
fi  
  
# enable programmable completion features (you don't need to enable  
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile  
# sources /etc/bash.bashrc).  
if ! shopt -oq posix; then  
    if [ -f /usr/share/bash-completion/bash_completion ]; then  
        . /usr/share/bash-completion/bash_completion  
    elif [ -f /etc/bash_completion ]; then  
        . /etc/bash_completion  
    fi  
fi  
export ARMLMD_LICENSE_FILE=8224@172.30.2.130
```

ARMLMD_LICENSE_FILE=license 文件的所在服务器的端口号@IP

- 2) 验证：

在 cmd 窗口中切换到 arm 编译器路径，输入编译器名称并回车。比如：armclang -help



```
peter@ubuntu: ~/ArmCompilerforEmbeddedFuSa6.16.2/bin  
peter@ubuntu:~/ArmCompilerforEmbeddedFuSa6.16.2/bin$ ls  
armar  armclang  armlm  fromelf  libLTO.so  suite_exec  
armasm  armlink  armlm-ipc  libarmlm.so  libstdc++.so.6  
peter@ubuntu:~/ArmCompilerforEmbeddedFuSa6.16.2/bin$ ./armclang --help  
Product:  
Component: Arm Compiler for Embedded FuSa 6.16.2  
Tool: armclang [5e007d00]  
  
Target: unspecified-arm-none-unspecified  
  
Usage: armclang --target=<triple> [options] file1 file2 ... fileN  
  
Mandatory options:  
--target          Generate code for the specified target triple. Supported values  
are:  
                  "arm-arm-none-eabi"      - generate code for AArch32 state  
                  "aarch64-arm-none-eabi" - generate code for AArch64 state  
  
Compiler and integrated assembler options:  
-march            Targets an architecture, generating generic code that runs on an  
y  
processor of that architecture.  
-mcpu             Targets a specific processor, generating optimized code for that  
specific processor.
```