

这俩软件主要在Ubuntu下进行安装。





将安装包拷贝到Ubuntu.



/\*

官方的.sh脚本无法下载，所以直接参考网上例程：

<https://blog.csdn.net/DongDong314/article/details/142797627?ops_request_misc=&request_id=&biz_id=102&utm_term=petalinux%E5%AE%98%E7%BD%91%E4%B8%8B%E8%BD%BD%E6%B5%81%E7%A8%8B&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-0-1>

使用命令来配置环境：

sudo apt-get install iproute2 gawk python3 python build-essential gcc git make net-tools libncurses5-dev tftpd zlib1g-dev libssl-dev flex bison libselinux1 gnupg wget git-core diffstat chrpath socat xterm autoconf libtool tar unzip texinfo zlib1g-dev gcc-multilib automake zlib1g:i386 screen pax gzip cpio python3-pip python3-pexpect xz-utils debianutils iputils-ping python3-git python3-jinja2 libegl1-mesa libsdl1.2-dev pylint3

\*/

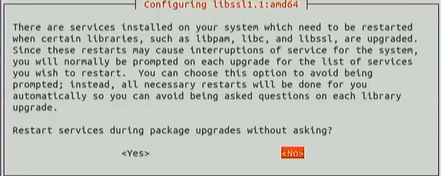
/\*

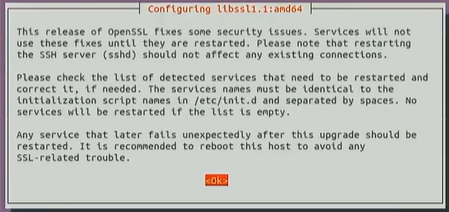
<https://support.xilinx.com/s/article/73296?language=en_US>



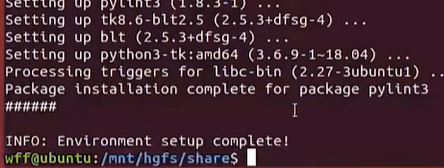
将这个脚本文件拷贝到虚拟机

Sudo执行该脚本文件。





安装完成后显示：



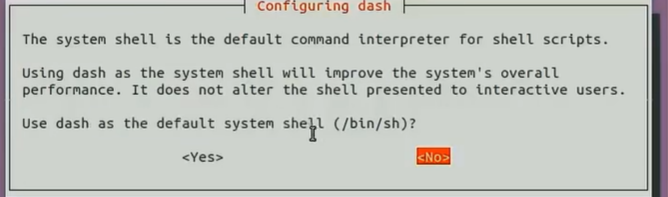
\*/

修改bash:  
 执行ls /bin/sh -alh

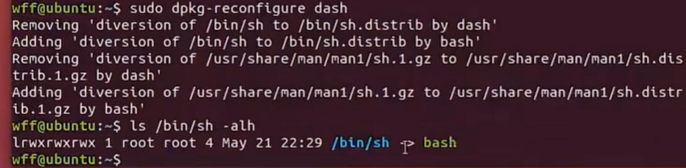
IMG_256

可见此时该连接文件指向dash，我们需要其指向bash

执行：sudo dpkg-reconfigure dash



再次执行ls /bin/sh -alh,此时指向的就是bash:



此时说明修改完成了，就需要安装了：

由于patalinux很大，所以建议安装在/opt/ 下面。

把opt目录当前属组的属组名 改成 当前用户名：

sudo chown -R $USER:$USER /opt

可以使用:

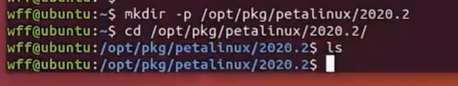
ls -alh 查看



在opt目录下创建一个文件夹，用于存放petalinux工具。

由于修改了属组，所以不需要加sudo的权限:

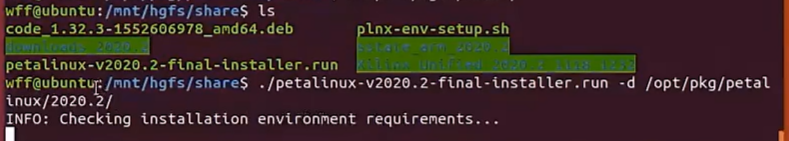
Mkdir -p /opt/pkg/petalinux/2020.2



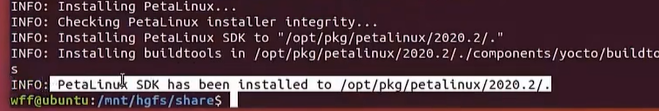
回到之前存放peta安装包的目录下，执行./petalinux安装peta软件

./petalinux + Tab补全 -d /opt/pkg/petalinux/2020.2/

（-d参数用于指定安装目录）



安装过程全程y同意，q退出。



/\*

25-2-22 13：23  
安装petalinux：

<https://blog.csdn.net/DongDong314/article/details/142797627?ops_request_misc=&request_id=&biz_id=102&utm_term=petalinux%E5%AE%98%E7%BD%91%E4%B8%8B%E8%BD%BD%E6%B5%81%E7%A8%8B&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-0-1>

参照上面的版本。

其次可以参考：

<https://blog.csdn.net/qq_37624854/article/details/137075661?ops_request_misc=%257B%2522request%255Fid%2522%253A%25224c85e0072311badccea3ba43d35f930c%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fall.%2522%257D&request_id=4c85e0072311badccea3ba4>

\*/

/\*

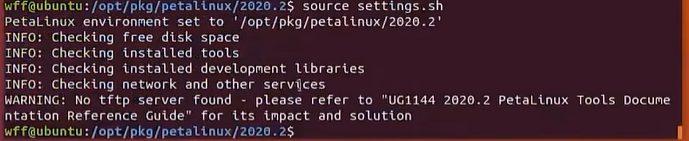
安装完成后，在/opt/pkg/petalinux/2020.2目录下会显示如下文件：

IMG_256

现在设置petalinux的环境变量：

在这个这个目录下可以看到有一个settings.sh和settings.csh。因为我们使用的是bash，所以就要使用.sh。

执行：source settings.sh



然后：

执行petalinux- 可看到如下信息，表示可以使用了：



上面的只是测试peta是否能正常运行，但另外打开一个终端又需要重新配置环境变量。所以可以设置一个变量的别名：

echo “alias sptl=’source /opt/pkg/petalinux/2020.2/settings.sh’” >> ~./bashrc

source ~/.bashrc

\*/

/\*

25-2-23 在安装完成时，进入到/opt/pkg/petalinux/2024.2/路径下，并没有他讲述的那些文件，而是：

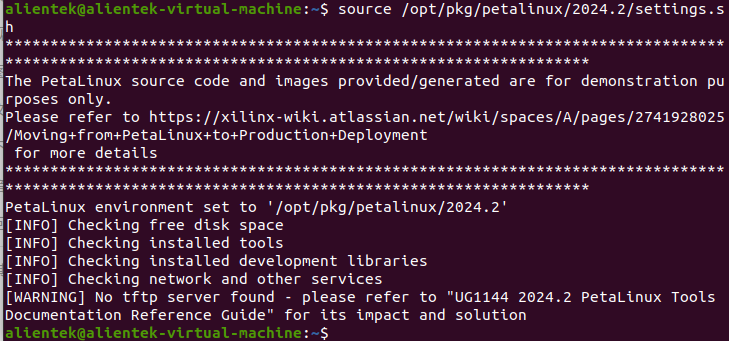
IMG_256

再次回到：

<https://blog.csdn.net/DongDong314/article/details/142797627?ops_request_misc=&request_id=&biz_id=102&utm_term=petalinux%E5%AE%98%E7%BD%91%E4%B8%8B%E8%BD%BD%E6%B5%81%E7%A8%8B&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-0-1>

看该引导的《3.使用示例》篇。

使用命令：source /opt/pkg/petalinux/2024.2/settings.sh可以正常看到：



验证工作环境是否已设置：使用命令：echo $PETALINUX可以看到：

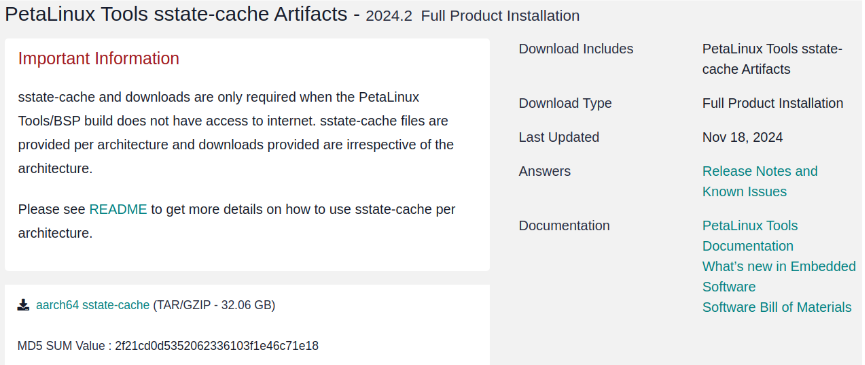
IMG_256

3.2 减少构建时间

Petalinux 工程编译时会从网上获取大量的必要资源，编译速度依赖于网速，如果网络不稳定还有可能在等待很久之后编译失败，推荐使用离线镜像。

离线镜像下载地址：

<https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools.html>



3.3 创建工程

>> petalinux-create project --template versal --name vd100

>> cd vd100 # 项目目录

参数解释如下：

--template <PLATFORM>， 以下平台类型受支持：

versal (for Versal adaptive SoC)

zynqMP (for Zynq UltraScale+ MPSoC)

zynq (for Zynq 7000 devices)

microblaze (for MicroBlaze processor)

注：MicroBlaze 模板不能用于Zynq 7000、Zynq MPSoC 或 Versal 的 PL 中。

--name <PROJECT\_NAME>，项目名称（项目目录）。

3.4 导入 XSA

petalinux-config --get-hw-description ../vd100\_ex\_pfm.xsa

3.5 基本配置

导入硬件时，会自动弹出 menuconfig 窗口（PetaLinux System Configuration）。

想要再次弹出此窗口，也可执行运行：

>> cd vd100 # 进入项目目录

>> petalinux-config

在此 menuconfig 中，进行一些基本的配置。

基本配置文件存储在：

<vd100\_prj>/project-spec/configs/config

\*/



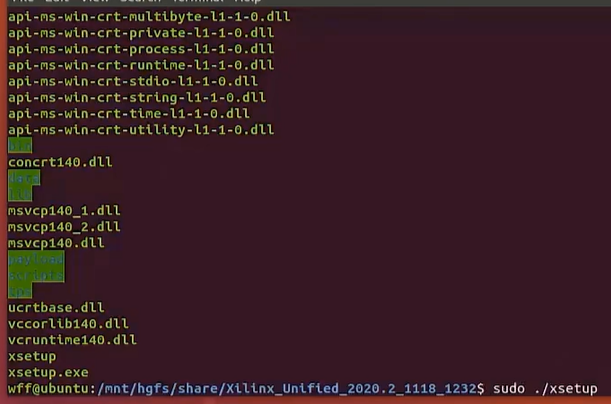
/\*

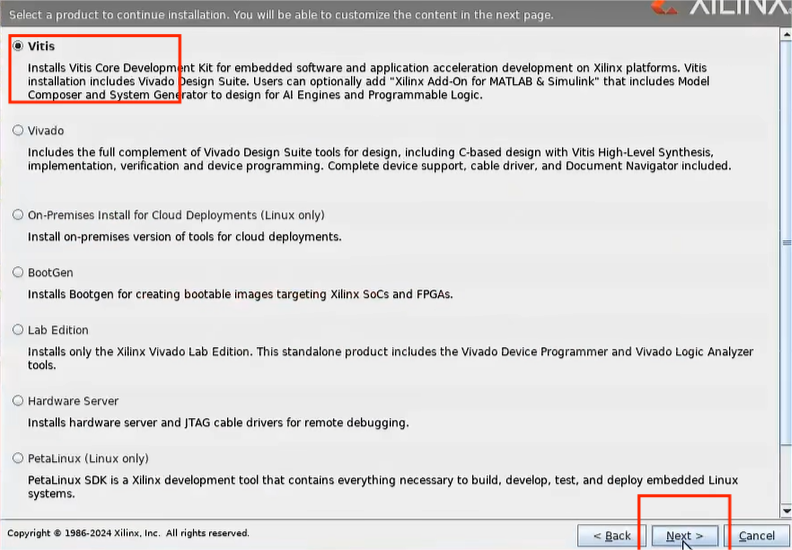
我们主要就是学习petalinux，所以暂不安装vitis（100多个G）。

\*/

将下载的软件包拷贝到虚拟机，安装vitis:

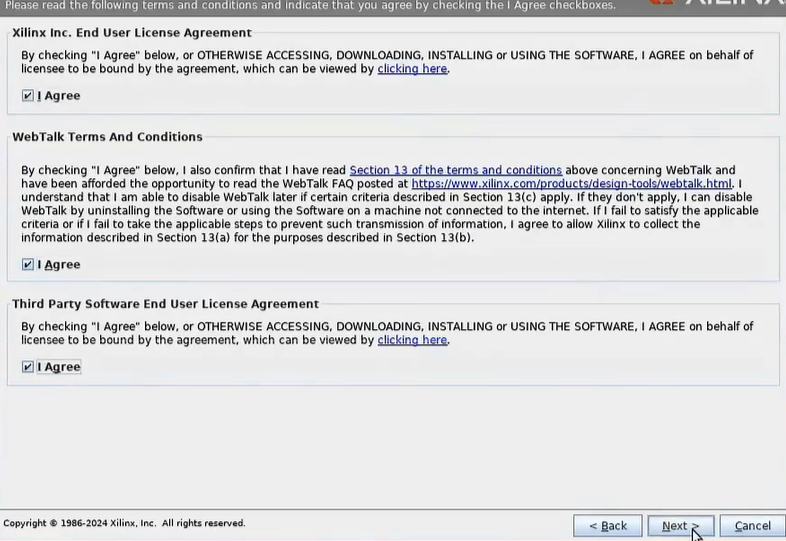
在软件包的目录下执行： sudo ./xsetup

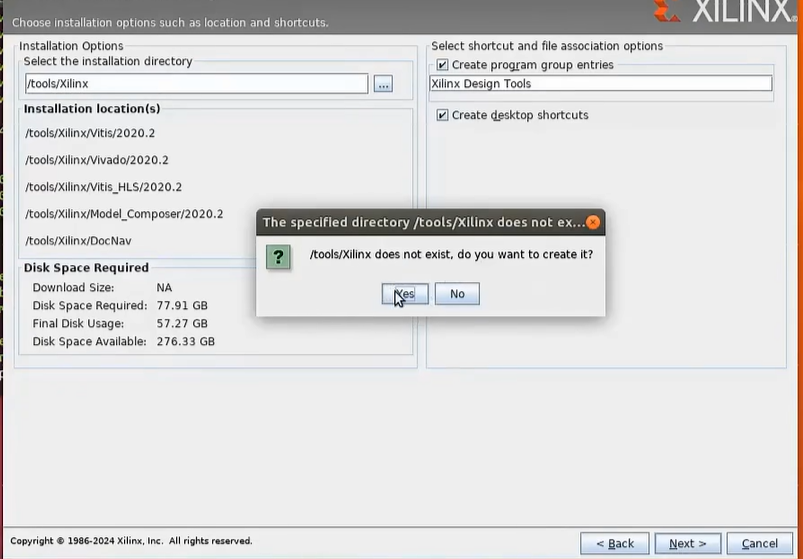


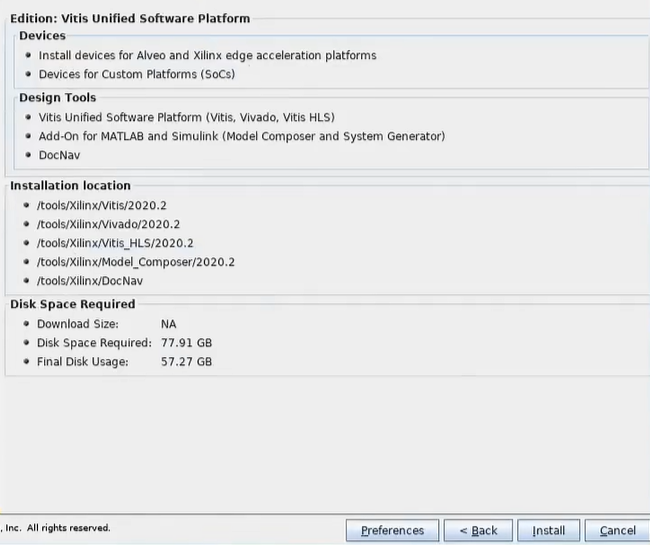




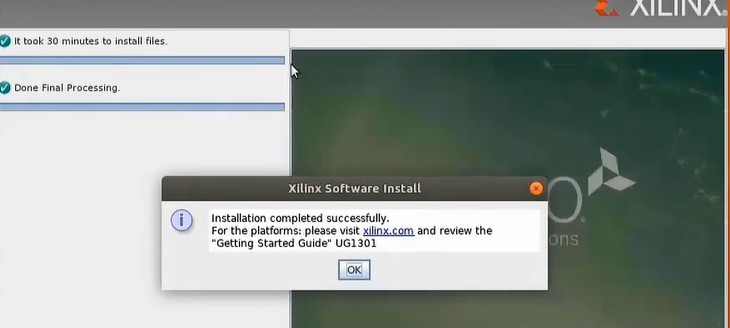
里面的RFSoc可以勾选，但空间会消耗更多。



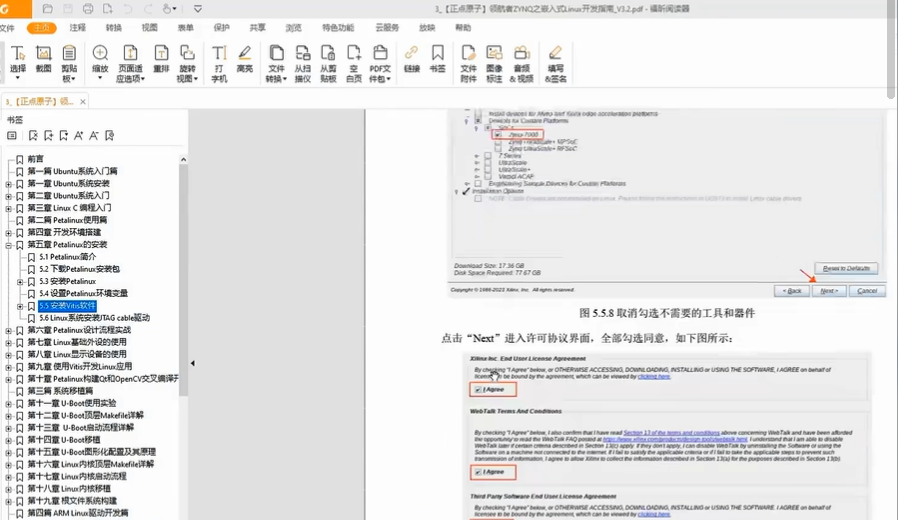


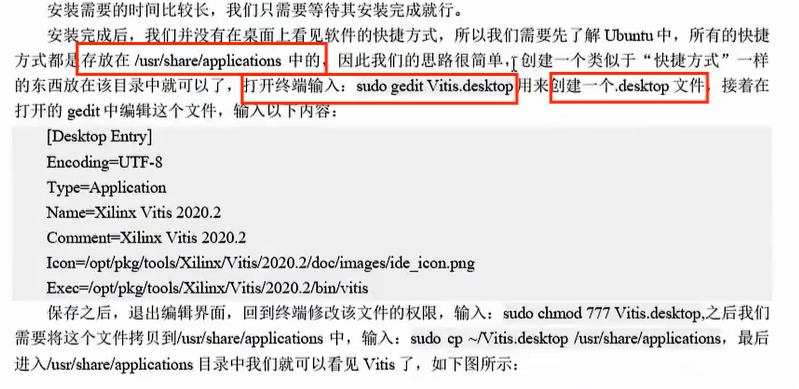


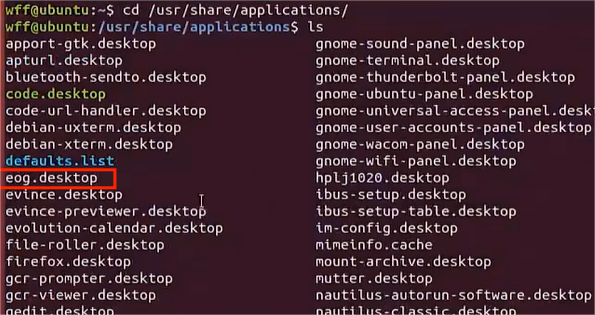
安装完成后显示：



由于没有快捷方式：后续vitis软件实操可以见领航者ZYNQ之嵌入式Linux开发指南手册







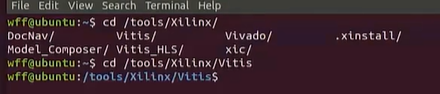
该目录下，这个软件就相当于是一个快捷方式。

sudo vim Vitis.desktop然后输入如下信息：

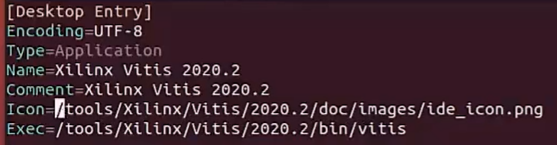


由于其vitis安装路径并不在上面Icon和Exec所示路径里面，所以需要修改：

Vitis安装路径：



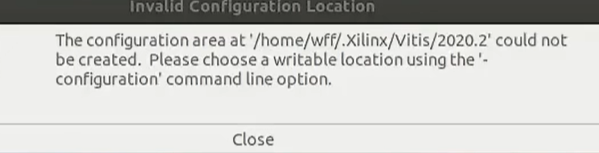
修改后的信息：



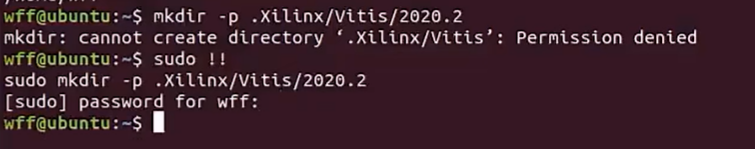
/\*

补充：

双击打开刚刚创建的图标。如果显示不能创建这个文件：



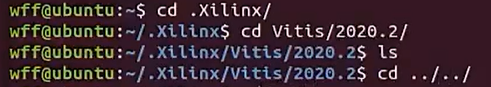
我们可以手动帮其创建（创建不了的原因有很多）



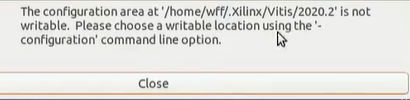
这里可以看到：创建不成功的原因就是权限不够。

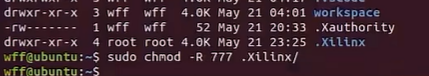
并且 “！！”表示执行上面（mkdir ...）的命令，相当于把上面的内容重复一下。

进到文件去看，可以看到确实创建成功了：



再次运行图标后如果显示：这个文件不可写：

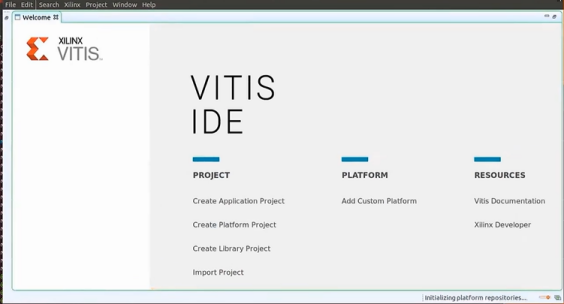




直接附加权限即可，这里-R表示递归（该目录下所有文件都执行这个命令）

\*/

双击打开后显示如下界面，表示vitis安装成功了：



/\*

Vitis百度网盘下载：

Vivado 2024.2

官方链接：

<https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/vivado-design-tools.html>

通过百度网盘分享的文件：Vivado2024.2

链接：

<https://pan.baidu.com/s/1QHp6tc277RL4CQjdxYJLOA?pwd=open>

提取码：open

\*/