WSL2部署chatGLM-6B @ Mobile-RTX3060

My github:https://github.com/alexhegit

OS环境准备（GPU+CUDA）：

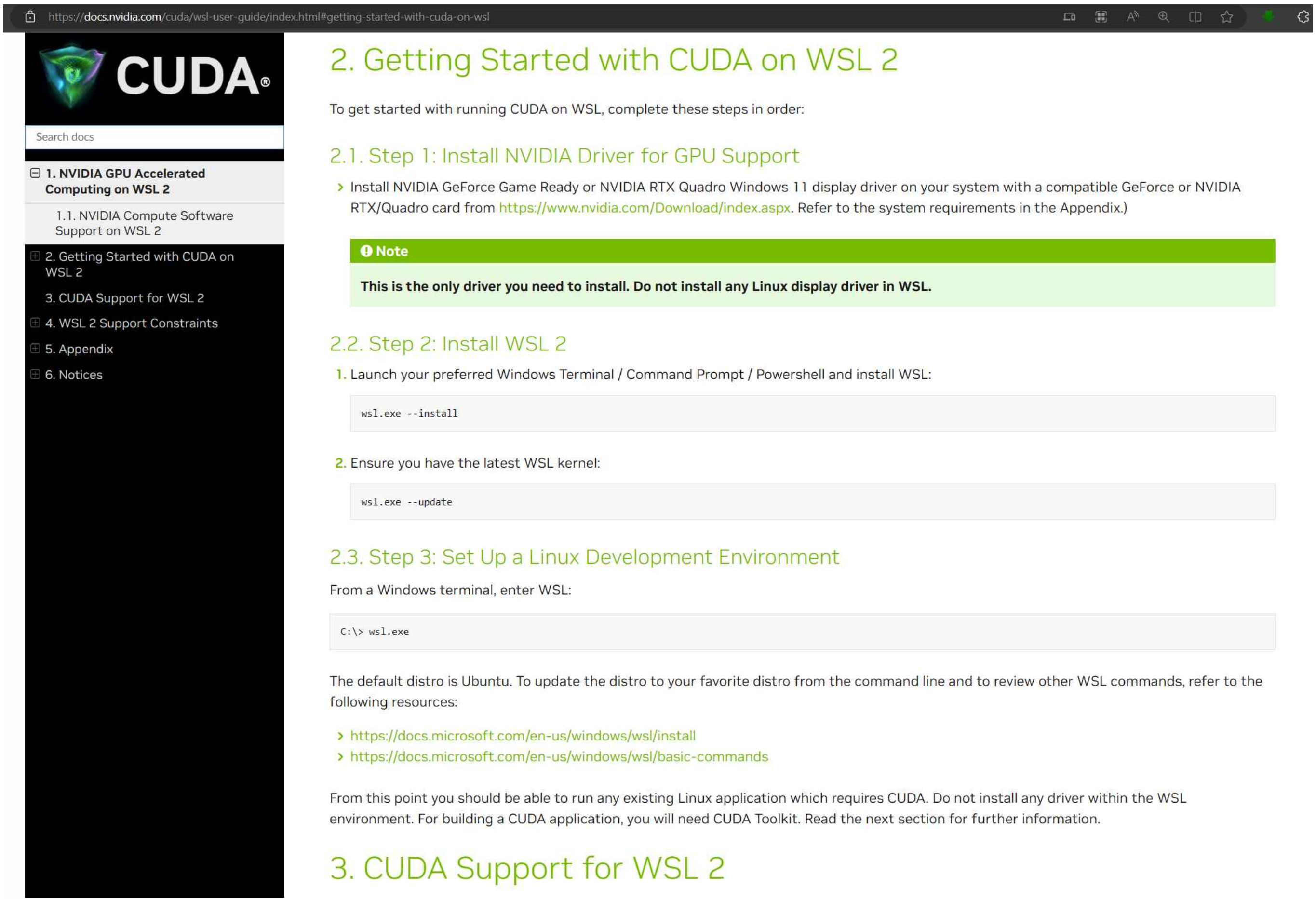
推荐两种OS环境

Option1: Windows11 + WSL2 + Ubuntu + CUDA

步骤:

1. 安装WSL2+Ubuntu
2. 在Windows11下安装GPU驱动，点击链接 https://www.nvidia.com/Download/index.aspx
3. 在WSL2中安装不带GPU驱动版本的CUDA Toolkit, 点击链接 downlad page for WSL-Ubuntu

> *点击展开获取以上步骤的**详尽过程（若以上安装无问题可以跳过）***
>
> *`这里假设您已经安装完WSL2+Ubuntu。下面需要安装为WSL2环境安装Driver及CUDA Toolkit。可参考：`*
>
> *`NVIDIA：https://docs.nvidia.com/cuda/wsl-user-guide/index.html#getting-started-with-cuda-on-wsl`*


https://docs.nvidia.com/cuda/wsl-user-guide/index.html#getting-started-with-cuda-on-wsl
CUDA
Search docs
1. NVIDIA GPU Accelerated Computing on WSL 2
1.1. NVIDIA Compute Software Support on WSL 2
2. Getting Started with CUDA on WSL 2
3. CUDA Support for WSL 2
4. WSL 2 Support Constraints
5. Appendix
6. Notices

2. Getting Started with CUDA on WSL 2

To get started with running CUDA on WSL, complete these steps in order:

2.1. Step 1: Install NVIDIA Driver for GPU Support

- Install NVIDIA GeForce Game Ready or NVIDIA RTX Quadro Windows 11 display driver on your system with a compatible GeForce or NVIDIA RTX/Quadro card from https://www.nvidia.com/Download/index.aspx. Refer to the system requirements in the Appendix.)

Note

This is the only driver you need to install. Do not install any Linux display driver in WSL.

2.2. Step 2: Install WSL 2

1. Launch your preferred Windows Terminal / Command Prompt / Powershell and install WSL:

```
wsl.exe --install
```

2. Ensure you have the latest WSL kernel:

```
wsl.exe --update
```

2.3. Step 3: Set Up a Linux Development Environment

From a Windows terminal, enter WSL:

```
C:\> wsl.exe
```

The default distro is Ubuntu. To update the distro to your favorite distro from the command line and to review other WSL commands, refer to the following resources:

- https://docs.microsoft.com/en-us/windows/wsl/install
- https://docs.microsoft.com/en-us/windows/wsl/basic-commands

From this point you should be able to run any existing Linux application which requires CUDA. Do not install any driver within the WSL environment. For building a CUDA application, you will need CUDA Toolkit. Read the next section for further information.

3. CUDA Support for WSL 2

这里特别提醒一下：

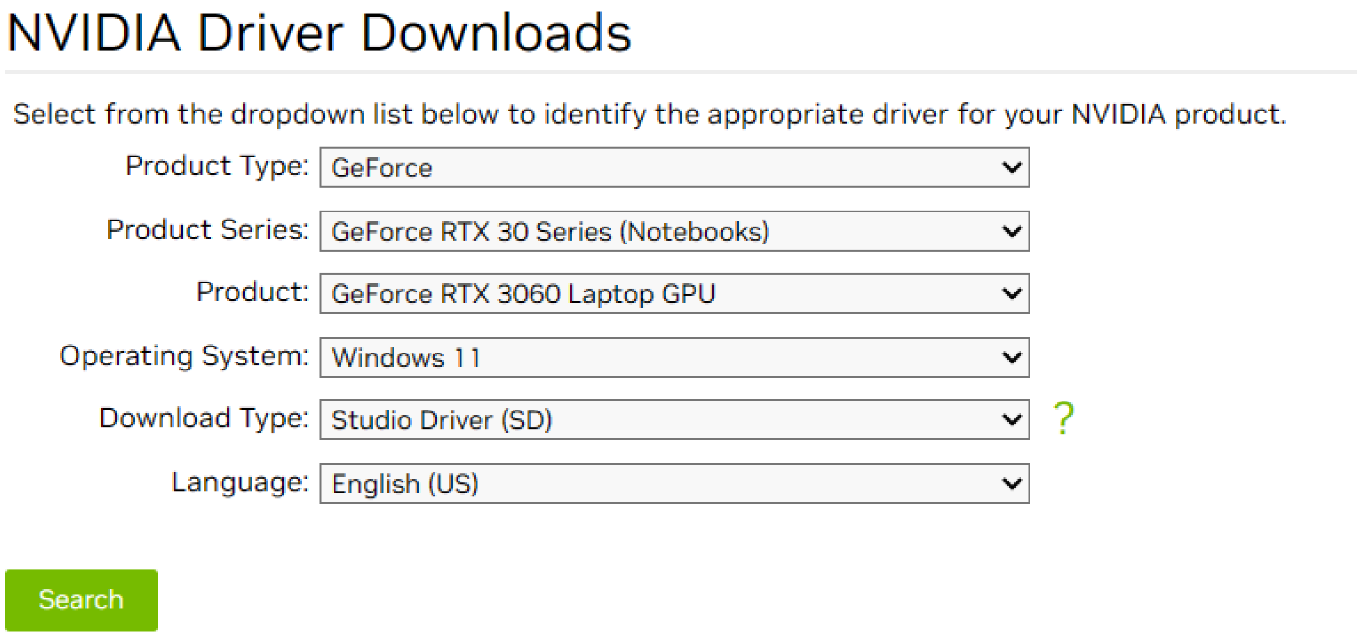
3. CUDA Support for WSL 2

The latest NVIDIA Windows GPU Driver will fully support WSL 2. With CUDA support in the driver, existing applications (compiled elsewhere on a Linux system for the same target GPU) can run unmodified within the WSL environment.

To compile new CUDA applications, a CUDA Toolkit for Linux x86 is needed. CUDA Toolkit support for WSL is still in preview stage as developer tools such as profilers are not available yet. However, CUDA application development is fully supported in the WSL2 environment, as a result, users should be able to compile new CUDA Linux applications with the latest CUDA Toolkit for x86 Linux.

Once a Windows NVIDIA GPU driver is installed on the system, CUDA becomes available within WSL 2. The CUDA driver installed on Windows host will be stubbed inside the WSL 2 as `libcuda.so`, therefore **users must not install any NVIDIA GPU Linux driver within WSL 2**. One has to be very careful here as the default CUDA Toolkit comes packaged with a driver, and it is easy to overwrite the WSL 2 NVIDIA driver with the default installation. We recommend developers to use a separate CUDA Toolkit for WSL 2 (Ubuntu) available from the CUDA Toolkit Downloads page to avoid this overwriting. This WSL-Ubuntu CUDA toolkit installer will not overwrite the NVIDIA driver that was already mapped into the WSL 2 environment. To learn how to compile CUDA applications, please read the CUDA documentation for Linux.

也就是说，在Windows中来安装CUDA Driver （https://www.nvidia.com/Download/index.aspx下载安装），该Driver以库文件libcuda.so的形式在WSL2中enable GPU。点击https://www.nvidia.com/Download/index.aspx，系统会自动探测机器中安装的GPU型号，如下图，点击Search后进入驱动下载后安装。



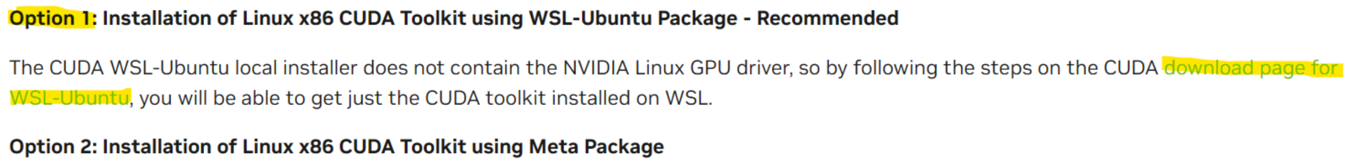
NVIDIA Driver Downloads

Select from the dropdown list below to identify the appropriate driver for your NVIDIA product.

Product Type: GeForce
Product Series: GeForce RTX 30 Series (Notebooks)
Product: GeForce RTX 3060 Laptop GPU
Operating System: Windows 11
Download Type: Studio Driver (SD)
Language: English (US)

然后，不要在WSL2+Ubuntu中以apt方式安装完整CUDA Toolkit，否则其中的Driver将覆盖之前在Windows环境下的驱动libcuda.so而导致无法在WSL2中使用GPU。

因此，需要在WSL2中安装不带Driver的CUDA Toolkit。我推荐使用方式一：

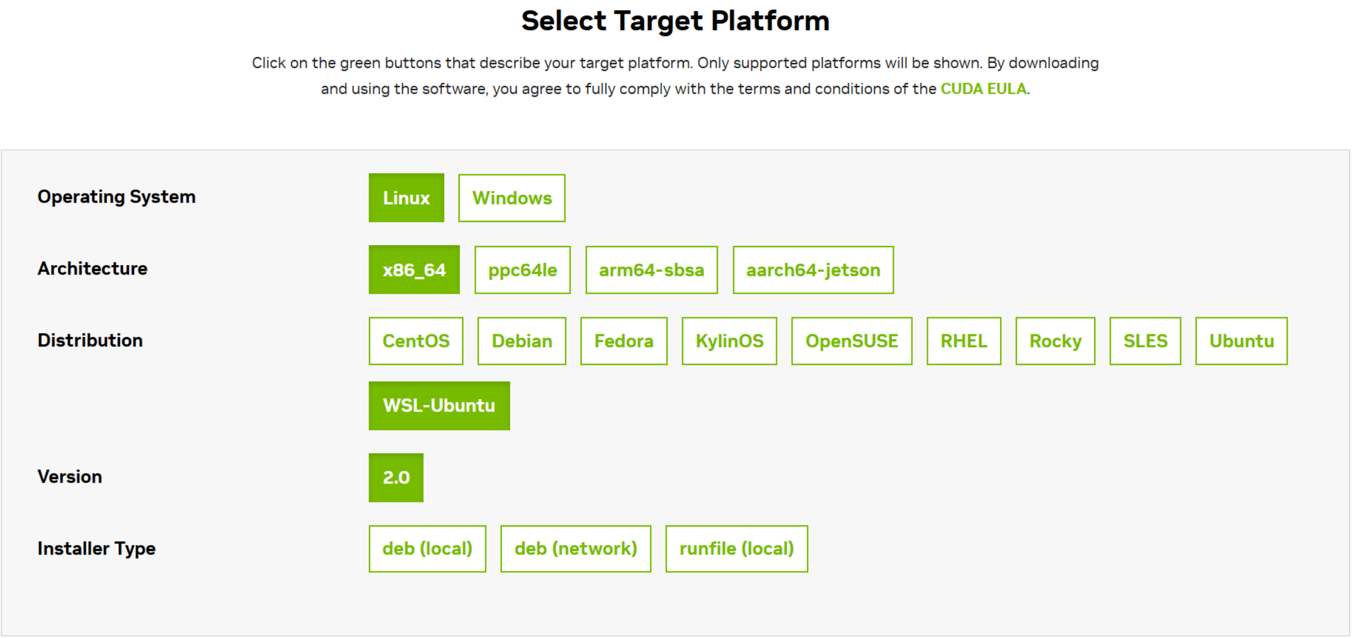


Option 1: Installation of Linux x86 CUDA Toolkit using WSL-Ubuntu Package - Recommended

The CUDA WSL-Ubuntu local installer does not contain the NVIDIA Linux GPU driver, so by following the steps on the CUDA download page for WSL-Ubuntu, you will be able to get just the CUDA toolkit installed on WSL.

Option 2: Installation of Linux x86 CUDA Toolkit using Meta Package

点击高亮的页面链接（download page for WSL-Ubuntu），进入安装选择页面。请依下图确认相关选择无误，最后一行的Installer Type可按个人喜好任意选择。



Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown. By downloading and using the software, you agree to fully comply with the terms and conditions of the CUDA EULA.

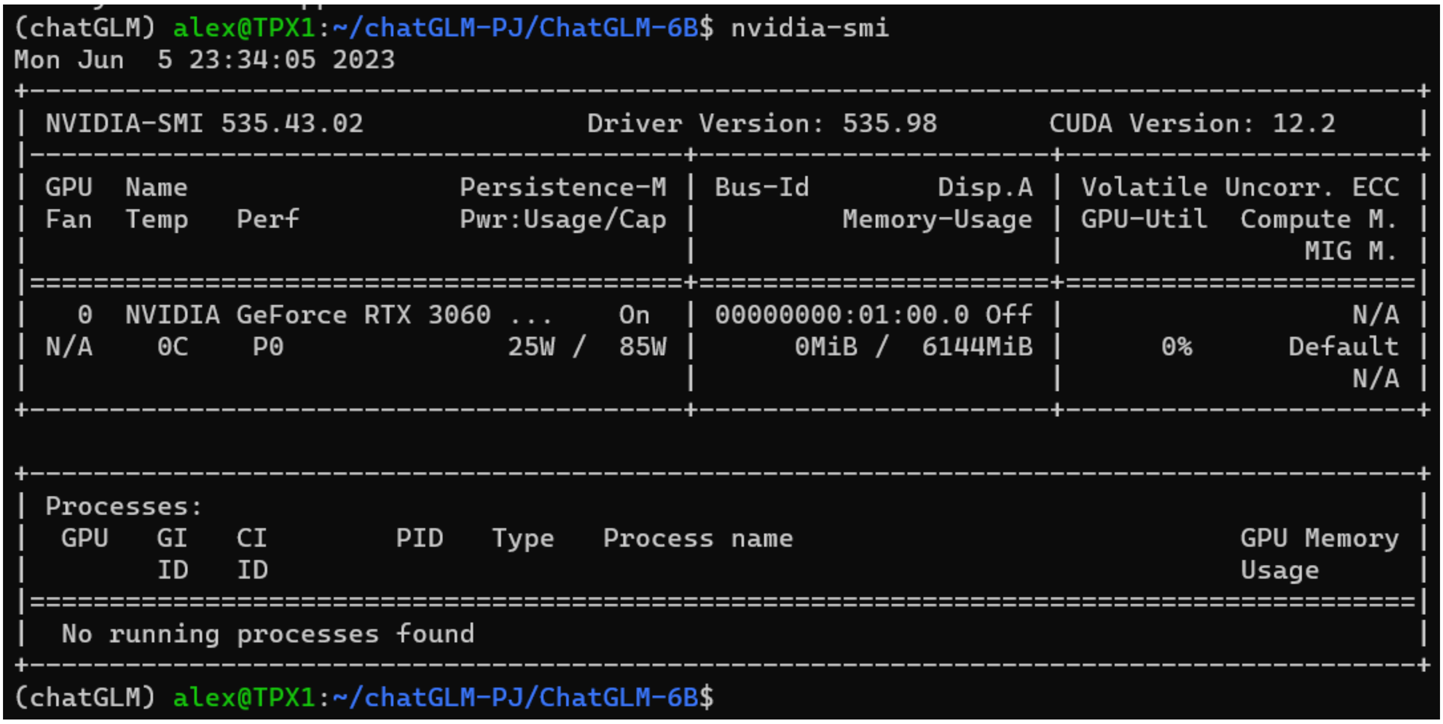
Operating System: Linux
Architecture: x86_64
Distribution: WSL-Ubuntu
Version: 2.0

Option2: Ubuntu20.04 + CUDA

在Ubuntu环境下准备CUDA环境比较容易，通过apt install相关CUDA toolkit/driver即可完成。

检查GPU+CUDA环境:

在WSL2或Ubuntu原生系统下，运行nvidia-smi。结果如图：



```
(chatGLM) alex@TPX1:~/chatGLM-PJ/ChatGLM-6B$ nvidia-smi
Mon Jun  5 23:34:05 2023
+---------------------------------------------------------------------------------------+
| NVIDIA-SMI 535.43.02              Driver Version: 535.98       CUDA Version: 12.2     |
|-----------------------------------------+----------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |         Memory-Usage | GPU-Util  Compute M. |
|                                         |                      |               MIG M. |
|=========================================+======================+======================|
|   0  NVIDIA GeForce RTX 3060 ...    On  | 00000000:01:00.0 Off |                  N/A |
| N/A    0C    P0              25W /  85W |      0MiB /  6144MiB |      0%      Default |
|                                         |                      |                  N/A |
+-----------------------------------------+----------------------+----------------------+

+---------------------------------------------------------------------------------------+
| Processes:                                                                            |
|  GPU   GI   CI        PID   Type   Process name                            GPU Memory |
|        ID   ID                                                             Usage      |
|=======================================================================================|
|  No running processes found                                                           |
+---------------------------------------------------------------------------------------+
(chatGLM) alex@TPX1:~/chatGLM-PJ/ChatGLM-6B$
```

下面进入主题来使用GPU部署chatGLM

部署chatGLM - Step by Step

1. 安装anaconda并初始化Python环境
    a. Install anaconda
    b. Create env

    ```
    conda  create -n chatGLM python=3.9
    ```

    ```
    conda activate chatGLM
    ```

2. Clone chatGLM repo

    ```
    git clone https://github.com/THUDM/ChatGLM-6B.git
    ```

    ```
    cd ChatGLM-6B
    ```

3. Install requirements from anaconda(env "chatGLM")

    ```
    pip install -r requirements.txt
    ```

> `$ cat requirements.txt`
>
> protobuf
>
> transformers==4.27.1
>
> cpm_kernels
>
> torch>=1.10
>
> gradio
>
> mdtex2html
>
> Sentencepiece

如果使用CUDA加速还需要安装（这是一个坑，官方文档中未提及）：

`conda install cudatoolkit`

4. Check PyTorch-CUDA

```
mkt@mkt-01:~/chatGLM-PJ/ChatGLM-6B$ python
Python 3.9.16 (main, Mar  8 2023, 14:00:05)
[GCC 11.2.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import torch
>>> print(torch.__version__)
2.0.1+cu117
>>> print(torch.cuda.is_available())
True
```

5. Download chatglm-6b-int4 pretrain model files

https://huggingface.co/THUDM/chatglm-6b-int4/tree/main

6. 运行simple_cli_demo.py

该脚本是基于huggingface中chatglm-6b-int4的Readme.md写的，可以做单次的对话用于测试。

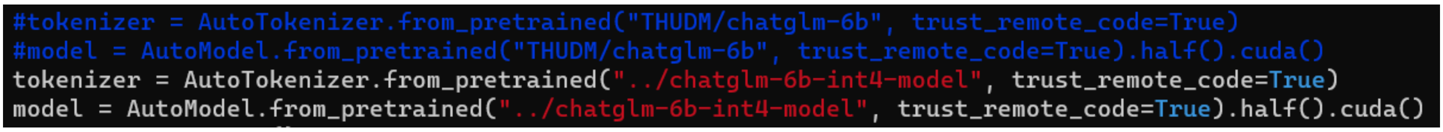
该脚本运行中将通过Huggingface下载chatglm-6b-int4量化模型文件。(https://huggingface.co/THUDM/chatglm-6b-int4)

```
from transformers import AutoTokenizer, AutoModel

tokenizer = AutoTokenizer.from_pretrained("THUDM/chatglm-6b-int4", trust_remote_
model = AutoModel.from_pretrained("THUDM/chatglm-6b-int4", trust_remote_code=Tru
model = model.eval()

response, history = model.chat(tokenizer, "你好", history=[])
```

```
print(response)

```

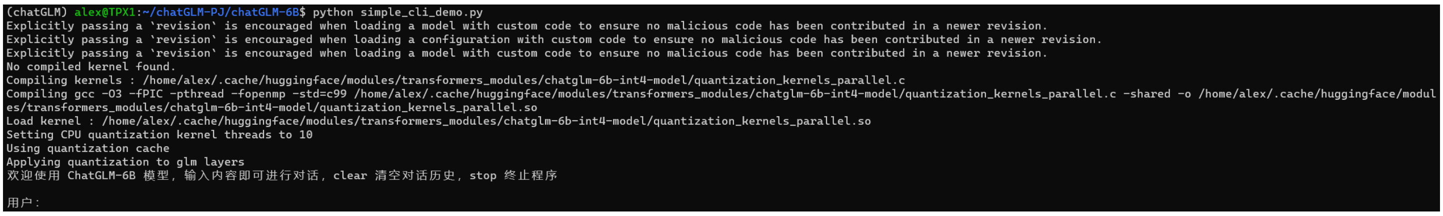
你也可以提前到https://huggingface.co/THUDM/chatglm-6b-int4下载所有文件到指定目录下，并修改上面的3，4两行代码中的模型路径到本地的模型文件存放目录的路径。

如果以上simple_cli_demo.py运行OK。那么可以修改cli_demo.py代码中的模型路径为本地存储的目录，来运行chatglm-yb-int4模型。修改如下图：



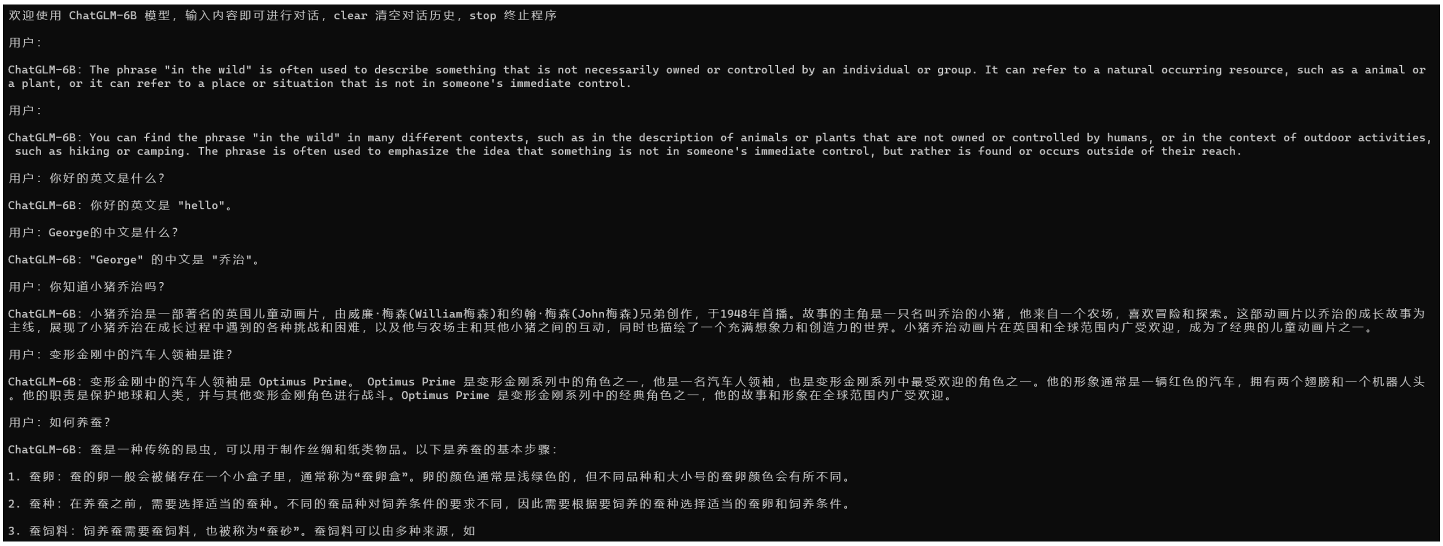
```
#tokenizer = AutoTokenizer.from_pretrained("THUDM/chatglm-6b", trust_remote_code=True)
#model = AutoModel.from_pretrained("THUDM/chatglm-6b", trust_remote_code=True).half().cuda()
tokenizer = AutoTokenizer.from_pretrained("../chatglm-6b-int4-model", trust_remote_code=True)
model = AutoModel.from_pretrained("../chatglm-6b-int4-model", trust_remote_code=True).half().cuda()
```

此后，就可以运行来进行多轮对话。启动过程如下图：



```
(chatGLM) alex@TPX1:~/chatGLM-PJ/chatGLM-6B$ python simple_cli_demo.py
Explicitly passing a `revision` is encouraged when loading a model with custom code to ensure no malicious code has been contributed in a newer revision.
Explicitly passing a `revision` is encouraged when loading a configuration with custom code to ensure no malicious code has been contributed in a newer revision.
Explicitly passing a `revision` is encouraged when loading a model with custom code to ensure no malicious code has been contributed in a newer revision.
No compiled kernel found.
Compiling kernels : /home/alex/.cache/huggingface/modules/transformers_modules/chatglm-6b-int4-model/quantization_kernels_parallel.c
Compiling gcc -O3 -fPIC -pthread -fopenmp -std=c99 /home/alex/.cache/huggingface/modules/transformers_modules/chatglm-6b-int4-model/quantization_kernels_parallel.c -shared -o /home/alex/.cache/huggingface/modul
es/transformers_modules/chatglm-6b-int4-model/quantization_kernels_parallel.so
Load kernel : /home/alex/.cache/huggingface/modules/transformers_modules/chatglm-6b-int4-model/quantization_kernels_parallel.so
Setting CPU quantization kernel threads to 10
Using quantization cache
Applying quantization to glm layers
欢迎使用 ChatGLM-6B 模型，输入内容即可进行对话，clear 清空对话历史，stop 终止程序

用户：
```

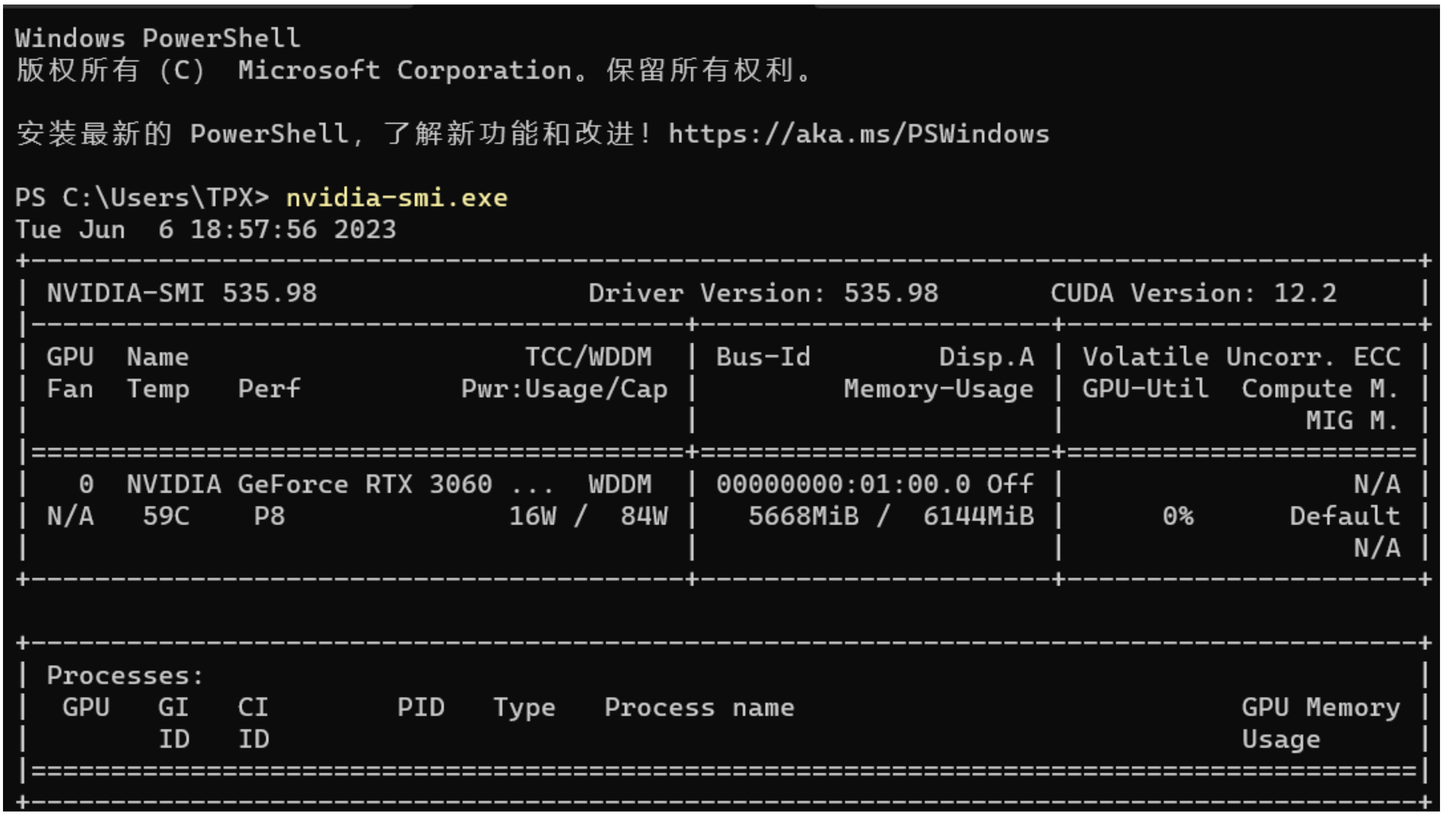
多轮对话示例：



```
欢迎使用 ChatGLM-6B 模型，输入内容即可进行对话，clear 清空对话历史，stop 终止程序

用户：

ChatGLM-6B: The phrase "in the wild" is often used to describe something that is not necessarily owned or controlled by an individual or group. It can refer to a natural occurring resource, such as a animal or
a plant, or it can refer to a place or situation that is not in someone's immediate control.

用户：

ChatGLM-6B: You can find the phrase "in the wild" in many different contexts, such as in the description of animals or plants that are not owned or controlled by humans, or in the context of outdoor activities,
 such as hiking or camping. The phrase is often used to emphasize the idea that something is not in someone's immediate control, but rather is found or occurs outside of their reach.

用户：你好的英文是什么？

ChatGLM-6B：你好的英文是 "hello"。

用户：George的中文是什么？

ChatGLM-6B："George" 的中文是 "乔治"。

用户：你知道小猪乔治吗？

ChatGLM-6B：小猪乔治是一部著名的英国儿童动画片，由威廉·梅森(William梅森)和约翰·梅森(John梅森)兄弟创作，于1948年首播。故事的主角是一只名叫乔治的小猪，他来自一个农场，喜欢冒险和探索。这部动画片以乔治的成长故事为
主线，展现了小猪乔治在成长过程中遇到的各种挑战和困难，以及他与农场主和其他小猪之间的互动，同时也描绘了一个充满想象力和创造力的世界。小猪乔治动画片在英国和全球范围内广受欢迎，成为了经典的儿童动画片之一。

用户：变形金刚中的汽车人领袖是谁？

ChatGLM-6B：变形金刚中的汽车人领袖是 Optimus Prime。 Optimus Prime 是变形金刚系列中的角色之一，他是一名汽车人领袖，也是变形金刚系列中最受欢迎的角色之一。他的形象通常是一辆红色的汽车，拥有两个翅膀和一个机器人头
。他的职责是保护地球和人类，并与其他变形金刚角色进行战斗。Optimus Prime 是变形金刚系列中的经典角色之一，他的故事和形象在全球范围内广受欢迎。

用户：如何养蚕？

ChatGLM-6B：蚕是一种传统的昆虫，可以用于制作丝绸和纸类物品。以下是养蚕的基本步骤：

1. 蚕卵：蚕的卵一般会被储存在一个小盒子里，通常称为“蚕卵盒”。卵的颜色通常是浅绿色的，但不同品种和大小号的蚕卵颜色会有所不同。

2. 蚕种：在养蚕之前，需要选择适当的蚕种。不同的蚕品种对饲养条件的要求不同，因此需要根据要饲养的蚕种选择适当的蚕卵和饲养条件。

3. 蚕饲料：饲养蚕需要蚕饲料，也被称为“蚕砂”。蚕饲料可以由多种来源，如
```

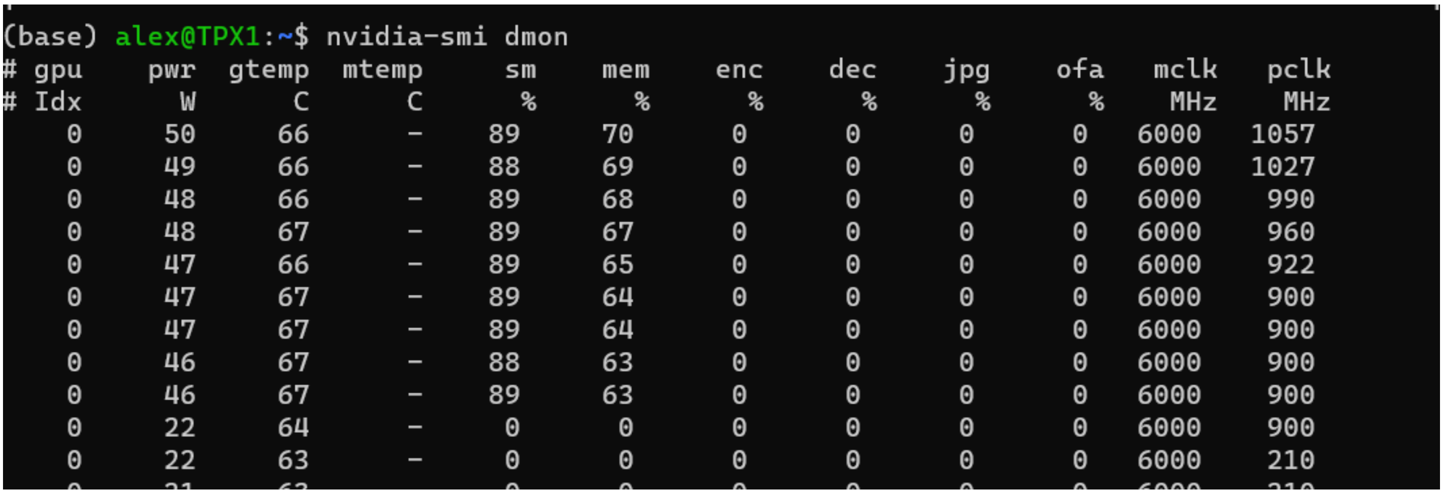
按Ctrl-Z可以中断退出。

同时使用nvidia-smi可以观察GPU的资源使用情况，可见int4的量化模型已经使用了接近6GB的GDDR空间。



```
Windows PowerShell
版权所有 (C)  Microsoft Corporation。保留所有权利。

安装最新的 PowerShell，了解新功能和改进！https://aka.ms/PSWindows

PS C:\Users\TPX> nvidia-smi.exe
Tue Jun  6 18:57:56 2023
+---------------------------------------------------------------------------------------+
| NVIDIA-SMI 535.98                 Driver Version: 535.98       CUDA Version: 12.2     |
|-----------------------------------------+----------------------+----------------------+
| GPU  Name                     TCC/WDDM  | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |         Memory-Usage | GPU-Util  Compute M. |
|                                         |                      |               MIG M. |
|=========================================+======================+======================|
|   0  NVIDIA GeForce RTX 3060 ...  WDDM  | 00000000:01:00.0 Off |                  N/A |
| N/A   59C    P8              16W /  84W |   5668MiB /  6144MiB |      0%      Default |
|                                         |                      |                  N/A |
+-----------------------------------------+----------------------+----------------------+

+---------------------------------------------------------------------------------------+
| Processes:                                                                            |
|  GPU   GI   CI        PID   Type   Process name                            GPU Memory |
|        ID   ID                                                             Usage      |
|=======================================================================================|
+---------------------------------------------------------------------------------------+
```

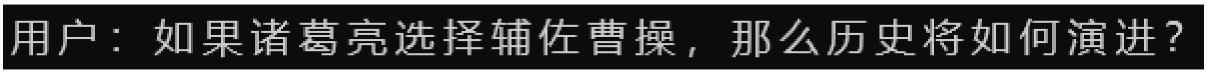
资源实时状态：



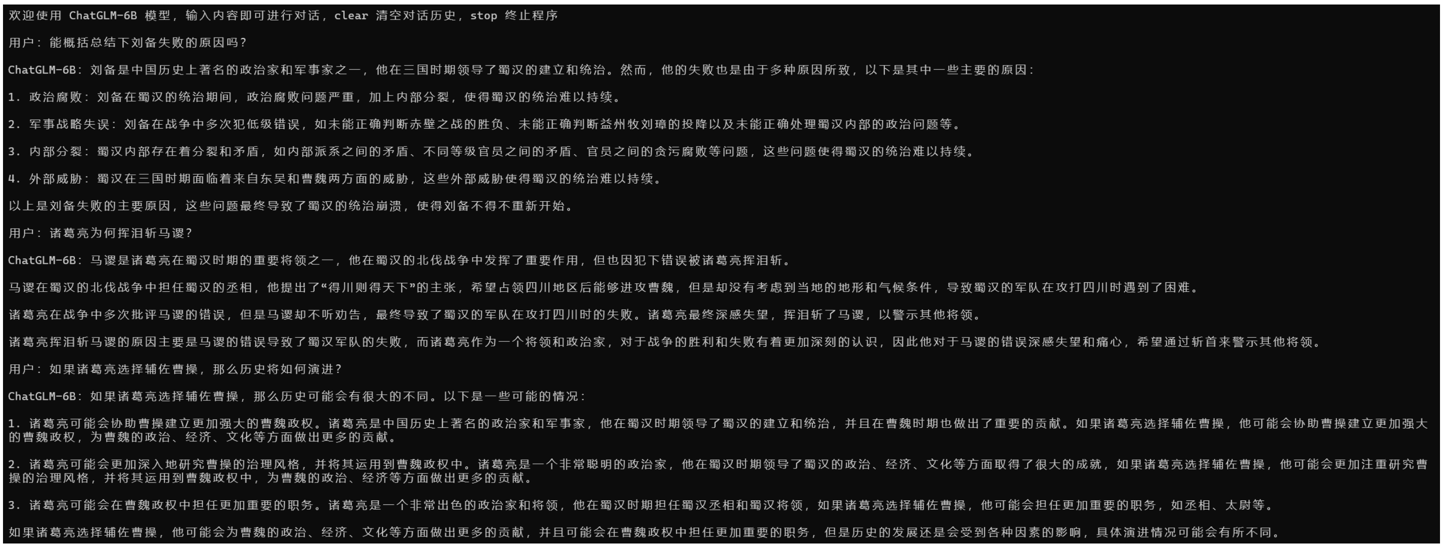
```
(base) alex@TPX1:~$ nvidia-smi dmon
# gpu    pwr  gtemp  mtemp     sm    mem    enc    dec    jpg    ofa   mclk   pclk
# Idx      W      C      C      %      %      %      %      %      %    MHz    MHz
    0     50     66      -     89     70      0      0      0      0   6000   1057
    0     49     66      -     88     69      0      0      0      0   6000   1027
    0     48     66      -     89     68      0      0      0      0   6000    990
    0     48     67      -     89     67      0      0      0      0   6000    960
    0     47     66      -     89     65      0      0      0      0   6000    922
    0     47     67      -     89     64      0      0      0      0   6000    900
    0     47     67      -     89     64      0      0      0      0   6000    900
    0     46     67      -     88     63      0      0      0      0   6000    900
    0     46     67      -     89     63      0      0      0      0   6000    900
    0     22     64      -      0      0      0      0      0      0   6000    900
    0     22     63      -      0      0      0      0      0      0   6000    210
```

目测下来，回答的汉字数可以达到每秒输出10个左右，但有时候速度会很慢，可能是问题比较难以回答（思考如何瞎编）。

比如如下图这个提问，它实在是编不下去（几分钟吐不出一个字）



用户：如果诸葛亮选择辅佐曹操，那么历史将如何演进？



```
欢迎使用 ChatGLM-6B 模型，输入内容即可进行对话，clear 清空对话历史，stop 终止程序

用户：能概括总结下刘备失败的原因吗？

ChatGLM-6B：刘备是中国历史上著名的政治家和军事家之一，他在三国时期领导了蜀汉的建立和统治。然而，他的失败也是由于多种原因所致，以下是其中一些主要的原因：

1. 政治腐败：刘备在蜀汉的统治期间，政治腐败问题严重，加上内部分裂，使得蜀汉的统治难以持续。

2. 军事战略失误：刘备在战争中多次犯低级错误，如未能正确判断赤壁之战的胜负、未能正确判断益州牧刘璋的投降以及未能正确处理蜀汉内部的政治问题等。

3. 内部分裂：蜀汉内部存在着分裂和矛盾，如内部派系之间的矛盾、不同等级官员之间的矛盾、官员之间的贪污腐败等问题，这些问题使得蜀汉的统治难以持续。

4. 外部威胁：蜀汉在三国时期面临着来自东吴和曹魏两方面的威胁，这些外部威胁使得蜀汉的统治难以持续。

以上是刘备失败的主要原因，这些问题最终导致了蜀汉的统治崩溃，使得刘备不得不重新开始。

用户：诸葛亮为何挥泪斩马谡？

ChatGLM-6B：马谡是诸葛亮在蜀汉时期的重要将领之一，他在蜀汉的北伐战争中发挥了重要作用，但也因犯下错误被诸葛亮挥泪斩。

马谡在蜀汉的北伐战争中担任蜀汉的丞相，他提出了“得川则得天下”的主张，希望占领四川地区后能够进攻曹魏，但是却没有考虑到当地的地形和气候条件，导致蜀汉的军队在攻打四川时遇到了困难。

诸葛亮在战争中多次批评马谡的错误，但是马谡却不听劝告，最终导致了蜀汉的军队在攻打四川时的失败。诸葛亮最终深感失望，挥泪斩了马谡，以警示其他将领。

诸葛亮挥泪斩马谡的原因主要是马谡的错误导致了蜀汉军队的失败，而诸葛亮作为一个将领和政治家，对于战争的胜利和失败有着更加深刻的认识，因此他对于马谡的错误深感失望和痛心，希望通过斩首来警示其他将领。

用户：如果诸葛亮选择辅佐曹操，那么历史将如何演进？

ChatGLM-6B：如果诸葛亮选择辅佐曹操，那么历史可能会有很大的不同。以下是一些可能的情况：

1. 诸葛亮可能会协助曹操建立更加强大的曹魏政权。诸葛亮是中国历史上著名的政治家和军事家，他在蜀汉时期领导了蜀汉的建立和统治，并且在曹魏时期也做出了重要的贡献。如果诸葛亮选择辅佐曹操，他可能会协助曹操建立更加强大
的曹魏政权，为曹魏的政治、经济、文化等方面做出更多的贡献。

2. 诸葛亮可能会更加深入地研究曹操的治理风格，并将其运用到曹魏政权中。诸葛亮是一个非常聪明的政治家，他在蜀汉时期领导了蜀汉的政治、经济、文化等方面取得了很大的成就，如果诸葛亮选择辅佐曹操，他可能会更加注重研究曹
操的治理风格，并将其运用到曹魏政权中，为曹魏的政治、经济等方面做出更多的贡献。

3. 诸葛亮可能会在曹魏政权中担任更加重要的职务。诸葛亮是一个非常出色的政治家和将领，他在蜀汉时期担任蜀汉丞相和蜀汉将领，如果诸葛亮选择辅佐曹操，他可能会担任更加重要的职务，如丞相、太尉等。

如果诸葛亮选择辅佐曹操，他可能会为曹魏的政治、经济、文化等方面做出更多的贡献，并且可能会在曹魏政权中担任更加重要的职务，但是历史的发展还是会受到各种因素的影响，具体演进情况可能会有所不同。
```

总之，个人离线搭建chatGLM-6b-int4这样的小模型在配置较低端GPU如RTX3060的笔记本上是具备一定可用性的。相信常规一点的知识问答是可以基本满足的。WSL2对CUDA的良好支持，避免了安装Windows/Linux双系统的一些麻烦。