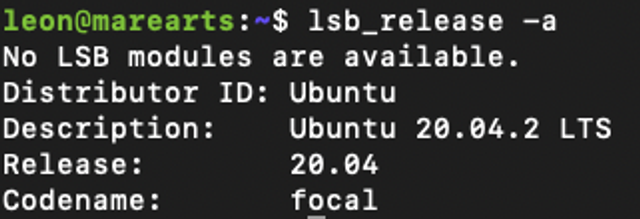
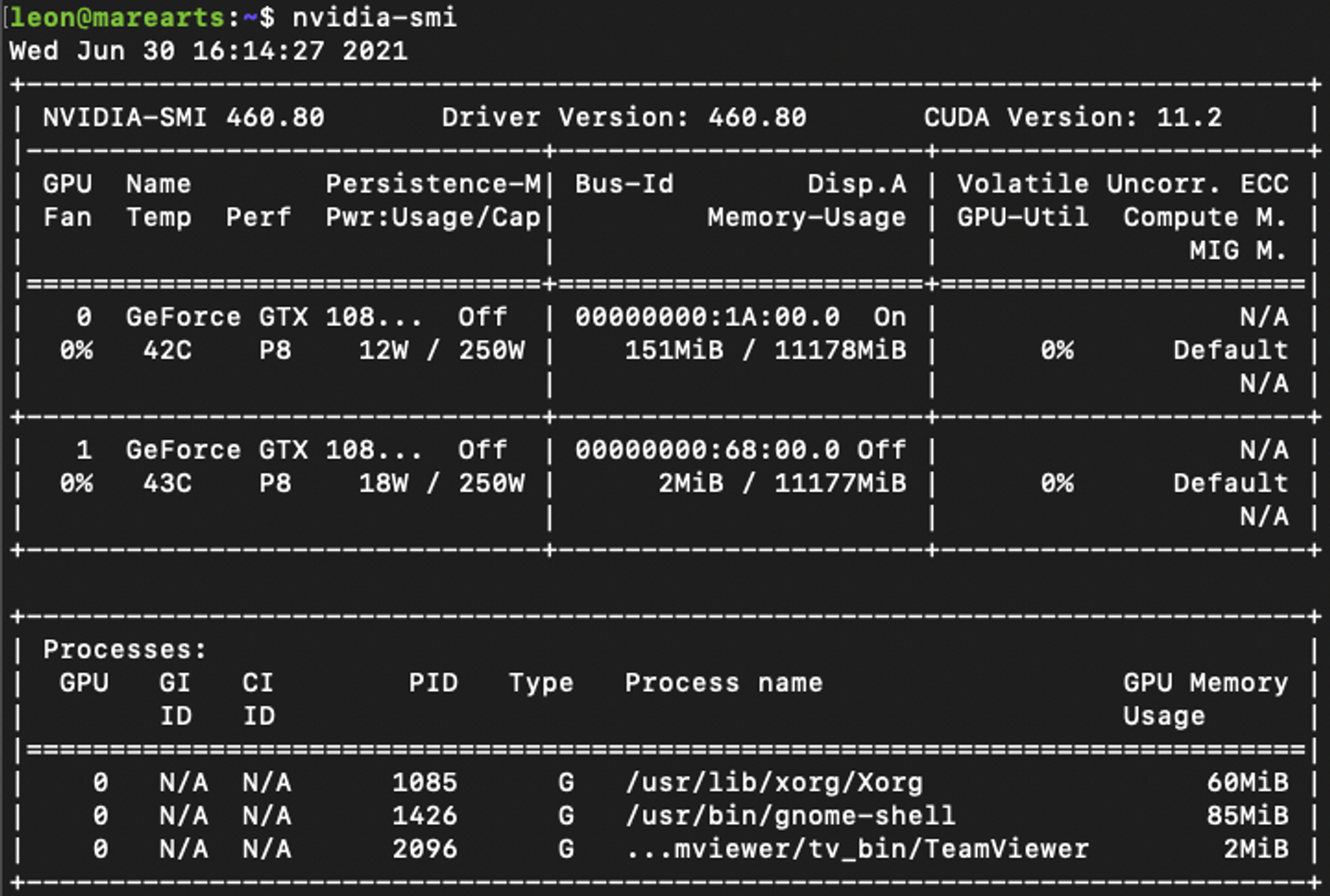
**1. Check your Ubuntu version**

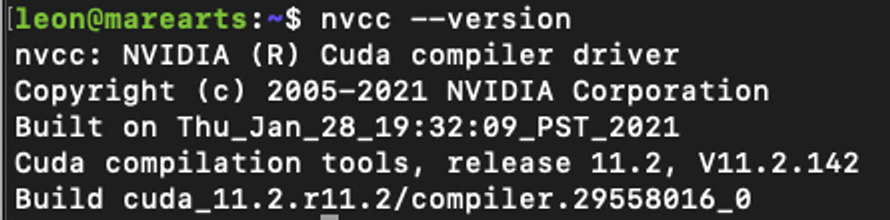
> lsb\_release -a****

**2. Check your CUDA, GPU driver is installed properly.**

> nvidia-smi



***> nvcc –version***



**2. Preparation**

**Install update and upgrade your system**

***> sudo apt update***

***> sudo apt upgrade***

**Install required libraries**

**# Generic tools:**

***> sudo apt install build-essential cmake pkg-config unzip yasm git checkinstall***

**# Image I/O libs**

***> sudo apt install libjpeg-dev libpng-dev libtiff-dev***

**# Video/Audio Libs - FFMPEG, GSTREAMER, x264 ans so on**

***> sudo apt install libavcodec-dev libavformat-dev libswscale-dev libavresample-dev***

***> sudo apt install libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev***

***> sudo apt install libxvidcore-dev x264 libx264-dev libfaac-dev libmp3lame-dev libtheora-dev***

***> sudo apt install libfaac-dev libmp3lame-dev libvorbis-dev***

**# OpenCore - Adaptive Multi Rate Narrow Band (AMRNB) and Wide Band (AMRWB) speech codec**

***> sudo apt install libopencore-amrnb-dev libopencore-amrwb-dev***

**# Cameras programming interface libs**

***> sudo apt-get install libdc1394-22 libdc1394-22-dev libxine2-dev libv4l-dev v4l-utils***

***> cd /usr/include/linux***

***> sudo ln -s -f ../libv4l1-videodev.h videodev.h***

***> cd ~***

**# GTK lib for the graphical user functionalities coming from OpenCV highghui module**

***> sudo apt-get install libgtk-3-dev***

**# Python libraries for python3:**

***> sudo apt-get install python3-dev python3-pip***

***> sudo -H pip3 install -U pip numpy***

***> sudo apt install python3-testresources***

**# Parallelism library C++ for CPU**

***> sudo apt-get install libtbb-dev***

**# Optimization libraries for OpenCV**

***> sudo apt-get install libatlas-base-dev gfortran***

**# Optional libraries:**

***> sudo apt-get install libprotobuf-dev protobuf-compiler***

***> sudo apt-get install libgoogle-glog-dev libgflags-dev***

***> sudo apt-get install libgphoto2-dev libeigen3-dev libhdf5-dev doxygen***

**2. OpenCV Build**

**# Install required libraries OpenCV**

***> cd ~/Downloads***

***> wget -O opencv.zip https://github.com/opencv/opencv/archive/refs/tags/4.5.5.zip***

***> wget -O opencv\_contrib.zip https://github.com/opencv/opencv\_contrib/archive/refs/tags/4.5.5.zip***

***> unzip opencv.zip***

***> unzip opencv\_contrib.zip***

***> cd opencv-4.5.5***

***> mkdir build***

***> cd build***

***> cmake -D CMAKE\_BUILD\_TYPE=RELEASE \***

***-D CMAKE\_INSTALL\_PREFIX=/usr/local \***

***-D WITH\_TBB=ON \***

***-D ENABLE\_FAST\_MATH=1 \***

***-D CUDA\_FAST\_MATH=1 \***

***-D WITH\_CUBLAS=1 \***

***-D WITH\_CUDA=ON \***

***-D BUILD\_opencv\_cudacodec=OFF \***

***-D WITH\_CUDNN=ON \***

***-D OPENCV\_DNN\_CUDA=ON \***

***-D CUDA\_ARCH\_BIN=7.5 \***

***-D WITH\_V4L=ON \***

***-D WITH\_QT=OFF \***

***-D WITH\_OPENGL=ON \***

***-D WITH\_GSTREAMER=ON \***

***-D OPENCV\_GENERATE\_PKGCONFIG=ON \***

***-D OPENCV\_PC\_FILE\_NAME=opencv.pc \***

***-D OPENCV\_ENABLE\_NONFREE=ON \***

***-D OPENCV\_PYTHON3\_INSTALL\_PATH=~/.virtualenvs/cv/lib/python3.8/site-packages \***

***-D PYTHON\_EXECUTABLE=~/.virtualenvs/cv/bin/python \***

***-D OPENCV\_EXTRA\_MODULES\_PATH=~/Downloads/opencv\_contrib-4.5.2/modules \***

***-D INSTALL\_PYTHON\_EXAMPLES=OFF \***

***-D INSTALL\_C\_EXAMPLES=OFF \***

***-D BUILD\_EXAMPLES=OFF \***

***-D CUDA\_ARCH\_BIN:STRING=6.0 6.1 7.0 7.5 \***

***-D BUILD\_opencv\_world=ON \***

***-D BUILD\_opencv\_contrib\_world=ON \***

***-D CUDA\_TOOLKIT\_ROOT\_DIR=/usr/local/cuda-11.4 ..***

***# Check how many cpu cores***

***> nproc***

***16***

***# build***

***> make -j16***

***# install all built libs into your system***

***> sudo make install***

***# include the libs in your environment***

***> sudo /bin/bash -c 'echo "/usr/local/lib" >> /etc/ld.so.conf.d/opencv.conf'***

***> sudo ldconfig***

**3.0 Test**

***#include <iostream>***

***#include <ctime>***

***#include <cmath>***

***#include "bits/time.h"***

***#include <opencv2/core.hpp>***

***#include <opencv2/highgui.hpp>***

***#include <opencv2/imgproc.hpp>***

***#include <opencv2/imgcodecs.hpp>***

***#include <opencv2/core/cuda.hpp>***

***#include <opencv2/cudaarithm.hpp>***

***#include <opencv2/cudaimgproc.hpp>***

***#define TestCUDA true***

***int main() {***

***std::clock\_t begin = std::clock();***

***try {***

***cv::String filename = "./example.jpg"; //change file name properly***

***cv::Mat srcHost = cv::imread(filename, cv::IMREAD\_GRAYSCALE);***

***for(int i=0; i<1000; i++) {***

***if(TestCUDA) {***

***cv::cuda::GpuMat dst, src;***

***src.upload(srcHost);***

***//cv::cuda::threshold(src,dst,128.0,255.0, CV\_THRESH\_BINARY);***

***cv::cuda::bilateralFilter(src,dst,3,1,1);***

***cv::Mat resultHost;***

***dst.download(resultHost);***

***} else {***

***cv::Mat dst;***

***cv::bilateralFilter(srcHost,dst,3,1,1);***

***}***

***}***

***//cv::imshow("Result",resultHost);***

***//cv::waitKey();***

***} catch(const cv::Exception& ex) {***

***std::cout << "Error: " << ex.what() << std::endl;***

***}***

***std::clock\_t end = std::clock();***

***std::cout << double(end-begin) / CLOCKS\_PER\_SEC << std::endl;***

***}***

**# Compile code & execute**

***> g++ test.cpp `pkg-config opencv --cflags --libs` -o test***

***> ./test***

**# Test if your system can find the OpenCV paths, run the following command**

***> pkg-config --cflags --libs opencv***

**# if it gives you an error message, your PKG\_CONFIG\_PATH environment variable may not be pointing to OpenCV and you must add the path on it. Try to run this and the test once again**

***> export PKG\_CONFIG\_PATH="$PKG\_CONFIG\_PATH:/usr/local/lib/pkgconfig"***

**# You can append the last command on a profile script to avoid the need to run it every time, try .bashrc:**

***> vi ~/.bashrc***