

CUDA_examples REPOSITORY TUTORIAL:

CUDA programs step by step tutorial for executing repository programs. It consist of tip dashes. Author strongly recommends book of J. Sanders et. al. "CUDA by example" and Youtube course by Udacity "intro to parallel programming".

1) obtain computer with included NVidia GPU graphics (programs are written in Nvidia corporation C extensions language with some C++11 capabilities - Compute Unified Device Architecture CUDA). If you have not any, I will provide you some new-parts proposition in 2017y:

- CPU: AMD A10 7860k / Intel G4560
- MB: Asrock FM2A88X+ (note PCIe 3.0 x16 bandwidth ~14GBps) / Asus H110M-K
- RAM: 2x GOODRAM PLAY Blue 8GB,
- pendrive 16GB with Ubuntu desktop 64bit, power supply,
- GPU: Nvidia GT730 4GB GDDR5 / GTX 1070 8GB
 - tip: consider operating system distribution booting to RAM (f.e. modified Ubuntu),
 - tip: consider good quality monitors, chair and keyboard (for my best knowledge mouse is unnecessary and slows work),
 - tip: for parallel cluster consider adding SSD RAID 0 (#mdadm are not so computationally expensive, as you think, especially for multicore CPU) and sufficient network interconnection,
 - tip: theoretical lower price – any used Personal Computer with PCIe 2.0x16 (f.e. Core2Quad + 2x2GB RAM + **GTX670** 2GB / **GTX780** 3GB). I did not check that configuration by myself, but should work (I have noticed non-compability of GPU with Dell R910, probably iDRAC issues),
 - tip: disabling graphical server (#sudo service lightdm stop) will give you additionally free space of +0.5GB RAM +0.34GB GPU memory, (author do recommend vim + screen in two windows), additionally some programs might execute faster,

2) further consideration are for LINUX UBUNTU (checked for 16.04). *I have been forced to check that on Microsoft Windows 10: Visual Studio Community (2015, great autoindent) and alternatively on Notepad++ with command line (shell) and both were working great (with different run commands).*

3) after signing in to operating system and making essential configurations (f.e. second sudo user for recovery mode) download CUDA toolkit from:

<https://developer.nvidia.com/cuda-toolkit>

4)to avoid:

- problems with lightdm log in (login loop)
- problems with driver istall ("Driver Installation failed: it appears, that a X server is running...")

via RUNFILE

5) and succesfully install a NVidia CUDA Toolkit on Ubuntu 16.04 64bit I have just had to do:

- login on live session on pendrive ("Try ubuntu, before install")
- add sudo user at live session:
 - #sudo adduser admin (#pass: admin1)
 - #sudo usermod -aG sudo admin
- logout from live session, log in as #admin
- download CUDA Toolkit from NVidia official site (~1.5GB)
- change privileges for downloaded installer file (DO NOT INSTALL AT THIS STEP!):
 - #sudo chmod +x cuda_X.X.run

- switch to console view:
 - #Ctrl+Alt+F1 (to switch on terminal view)
 - #Ctrl+Alt+F7 (to switch from terminal view to graphical server)
- at console view (Ctrl+Alt+F1) log in:
 - #login: admin
 - #pass: admin1
- stop graphical running service:
 - #sudo service lightdm stop
- check if graphical server is off - after switching Ctrl+Alt+F7 the monitor should be blank black, switch back on console view Ctrl+Alt+F1
- install CUDA Toolkit, with such configuration:
 - #sudo ./cuda_X.X.run
 - ##(press 'q' for license read skip)
 - #do not install OpenGL library (I do not know why – please do not ask)
 - #do not update system X configuration
 - #other options make yes and paths as default
- turn on graphical server:
 - #sudo service lightdm start
- log in as user (if you automatically log in as #ubuntu at live session log out):
 - #login: admin
 - #pass: admin1
- check if nvcc exists:
 - # sudo find /usr/ -name 'nvcc'

6) obtain git:

#sudo apt-get install git

7) clone my repo in home directory (~):

#git clone https://github.com/PiotrLenarczykAnonim/CUDA_examples.git

8) check whatever nvcc compiler works:

#cd CUDA_examples/01_makeSimple/

9) most of folders are configured for #make via BASH scripts:

#./RUN_COMMANDS.sh

10) Thrust library is delivered with CUDA Toolkit, but I strongly recommend for cloning repo Mr Jared Hoberock with examples:

git clone <https://github.com/thrust/thrust.git>

11) there are another repo with C++ several useful examples:

#git clone https://github.com/PiotrLenarczykAnonim/C-_examples.git

via .DEB PACKAGE

5deb) download download cudaXX.deb package from

<https://developer.nvidia.com/cuda-toolkit>

6deb) in console:

- switch to console view:
 - #Ctrl+Alt+F1 (to switch on terminal view)
 - #Ctrl+Alt+F7 (to switch from terminal view to graphical server)
- at console view (Ctrl+Alt+F1) log in:
 - #login: admin
 - #pass: admin1
- stop graphical running service:
 - #sudo service lightdm stop
- check if graphical server is off - after switching Ctrl+Alt+F7 the monitor should be blank black, switch back on console view Ctrl+Alt+F1
 - #sudo chmod +x cudaXX.deb
 - #sudo dpkg -i cudaXX.deb

```
#sudo apt-get update
#sudo aptitude install cuda g++-4.9 gcc-4.9
#sudo ln -s /usr/bin/g++-4.9 /usr/local/cuda/bin/g++
#sudo ln -s /usr/bin/gcc-4.9 /usr/local/cuda/bin/gcc
#sudo service lightdm start
```

7deb) similar to above 6) - 11)

Post Scriptum: Mostly I have been developing it on Dell Inspiron 7746 with i7 5500U, 16GB RAM 1666MHz, Nvidia GM108M GeForce 845M PCIe 3.0x4 2GB and SSD 1TB (OS: LINUX Ubuntu Desktop 16.04 64bit; LINUX Ubuntu Desktop 16.10 64bit). Also it was checked on PC: Intel i7 2660k, 16GB RAM 1333MHz, Nvidia GK110 GTX780 3GB PCIe 2.0x16, SSD 0.24TB (OS: LINUX Ubuntu 16.04 64bit).

Post Post Scriptum: If one will use customized liveCD from LINUX_tips repo, he or she should take into consideration that minimal recommended RAM size should be 8GB (or buy harddrive / pendrive).