CUSTOM LINUX LIVECD TUTORIAL

1) making custom LiveCD booting and entirely running in RAM memory one could find quite fast and useful. This distro is configured for M-script (MATLAB / Octave) and C++ programming issues. It consist of removing packages; installing a pack of different programms; deleting welcome Try/Install window; repository updates. Minimal requirements:

- any modern: motherboard, processor, power suply and RAM memory
- author recommends ~4GB of RAM for typical basic scientific usage (0.9-1.1 GB consumes Ubuntu Desktop 16.10 operating system)
- PC's with lower amount of RAM should boot with other solution like Puppy LINUX
- a HDD is useful anyway for results backup – pendrive should do the job

2) to start from scratch one needs LINUX running enviroment (this tutorial was built on Ubuntu 16.04 LTS and produced a modified liveCD Ubuntu 16.10 64bit Desktop at 15.03.2017y.) and LINUX distribution ISO file (~1.6GB) downloaded from:

https://www.ubuntu.com/download/desktop

3) download and start customizing program Cubic on host OS:

```
#sudo apt-add-repository ppa:cubic-wizard/release
#sudo apt-get update
#sudo apt-get install cubic
#cubic
```

4) after choosing custom liveCD project folder and downloaded ISO file, Cubic starts a customized client OS console (default running as root user #sudo su) for further liveCD operating system changes. Firstly one should upgrade repo's – author uses those polish mirror repos listed below:

```
#nano /etc/apt/sources.list      #( delete content )
deb http://pl.archive.ubuntu.com/ubuntu/ xenial main restricted
deb http://pl.archive.ubuntu.com/ubuntu/ xenial-updates main restricted
deb http://pl.archive.ubuntu.com/ubuntu/ xenial universe
deb http://pl.archive.ubuntu.com/ubuntu/ xenial-updates universe
deb http://pl.archive.ubuntu.com/ubuntu/ xenial multiverse
deb http://pl.archive.ubuntu.com/ubuntu/ xenial-updates multiverse
deb http://pl.archive.ubuntu.com/ubuntu/ xenial-backports main restricted universe multiverse
deb http://security.ubuntu.com/ubuntu xenial-security main restricted
deb http://security.ubuntu.com/ubuntu xenial-security universe
deb http://security.ubuntu.com/ubuntu xenial-security multiverse
#add-apt-repository universe
#add-apt-repository main
#add-apt-repository restricted
#add-apt-repository multiverse
```

5) remove office, email, full system install packages (~770MB):

```
#apt-get remove libreoffice-*
#apt-get remove thunderbir*
#apt-get remove ubiquit*
```

6) update repos:

```
#apt-get update && apt-get upgrade && apt-get update && apt-get autoremove
```

7) install packages (Qt, Octave, editors, git, VNC server+client, link aggregation, samba ~900MB):

```
#apt-get install aptitude
#aptitude install htop w3m gpm vim git konsole screen kate parallel p7zip-full mc qt5-default cppman vlc libvncserver-dev tightvncserver xtightvncviewer xfce4 xfce4-goodies octave octave-communications octave-communications-common octave-control octave-image octave-signal doxygen iperf ifenslave samba cifs-utils openssh-server traceroute
```

8) drag & drop needed files and directories to Cubic console

9) customize vim (after „syntax on"):

```
#nano /etc/vim/vimrc
```

set autoindent
set ts=4
set sw=4
set mouse=a
set number

10) customize grub:

#vim /etc/default/grub

#GRUB_HIDDEN_TIMEOUT_QUIET=true
#GRUB_TIMEOUT=10
GRUB_CMDLINE_LINUX_DEFAULT="toram"

#update-grub

11) allow remote ssh login:

#sudo ufw allow 22

12) set polish language („pl_PL.UTF-8”) as default:

#apt-get install language-pack-pl-base language-pack-pl
#dpkg-reconfigure locales

13) start a boot disc creator for pendrive and choose a customized ISO file, restart, change boot options in BIOS and voile:)

14) customized liveCD has ~1.8GB in size (+0.2GB comparing to original Ubuntu Desktop)

15) one can add some further additional features to liveCD using customized ISO file as input to Cubic package,

16) additionaly if one will be making Ubuntu Desktop instalation with ordinary live pendrive and UEFI BIOS (legacy will be deprecated in my best knowledge), he or she should make such partitions structure of disk:

-EFI partition ~100MB,
-swap partition ~120% of RAM size,
-free space as ext4 filesystem and root „/” directory as bootable.

Moreover UEFI bootable script should be choosen from EFI partition and „grub.cfg” file (filenames could vary).