



Estonian Information
Technology College

Software management

Operating systems 1800

Edmund Laugasson
edmund.laugasson@itcollege.ee

There has been used materials from Margus Ernits, Katrin Loodus when creating current slides.

Current document copying, distributing and/or modifying has been set out by one of the following licences by user's choice:

* GNU Free Documentation Licence version 1.2 or newer

* Creative Commons Attribution + ShareAlike licence 4.0 (CC BY-SA)

Software distribution

- software is distributed
 - as binaries
 - as software packages, e.g. msi, rpm, deb etc
 - intermediate code (partially compiled or object code) e.g. jar, class, .net assembly etc

101011001....
 - as source-code
 - scripting languages python, php, perl, ruby etc
 - *source-code*, which has to be compiled into binary or intermediate code before use
 - on the media and in the network

The software is divided into

- Software can be divided based on management
 - system software
 - drivers
 - operating system modules and main components
 - application software
 - services
 - user tools
- Properly planned application works and can be installed with user rights

Problem

- Pieces of software tend to depend on other libraries
 - It makes no sense to do everything myself
- Libraries and dependencies solving is often left to the user's own responsibility
 - DLL Hell - partially solved
 - lib32, lib64
 - Dynamically Linked Libraries vs. statically linked libraries
- Some of the software requires Administrator / root privileges to start up

Good software installation mechanism

- Will be able to resolve dependencies
 - Installing a package that requires a second, then will be found from the network and installed automatically
- Allows you to search for software
 - this would be very annoying and not safe (*adware*) to search software through the Internet
- tells about the package
 - for each package, you can ask
 - status, e.g. `dpkg -s nano`
 - search for a filename from installed packages, e.g. `dpkg -S nano`
 - list files installed to your system, e.g. `dpkg -L nano`
- Allows list of software to clone to a new machine
- Allows you to check the authenticity of the origin of software

Software package in Linux systems

- Linux-like operating systems have generally in the software package management system
- Software packages are digitally signed (*PGP-signed*)
- Software management system keeps information about the installed software packages
 - Name and description
 - Version
 - List of files in package
 - Dependencies

<https://wiki.debian.org/SecureApt>

<https://help.ubuntu.com/community/SecureApt>

Software package dependencies

- package can depend on other packages
 - e.g. web browser depends on X Window System libraries
- package can conflict with any other package that has to be resolved e.g. by removing another package
- some packages are equivalent (e.g. some package may depend from one or another – then both are not needed)

Package management systems

- for software management there is often used to package management systems
 - rpm (file type .rpm)
 - apt and dpkg (file type .deb)
 - yum, dnf (file type .rpm)
 - etc
- Package management systems takes care of different things:
 - Working with file archivers to extract package archives
 - Ensuring the integrity and authenticity of the package by verifying their digital certificates and checksums
 - Looking up, downloading, installing or updating existing software from a software repository or app store
 - Grouping packages by function to reduce user confusion
 - Managing dependencies to ensure a package is installed with all packages it requires, thus avoiding "dependency hell"

Software installation in Ubuntu

- to install software you have to know exact package name (can be searched beforehand)
 - ***sudo apt-get install package***
 - ***sudo apt-get install firefox*** (see also man apt-get)
 - ***sudo apt install firefox*** (see also man apt)
- when installing then there will be checked dependencies written in package header and also these will be installed
- package installation can be also simulated to ensure that nothing will be broken:
 - ***sudo apt-get install firefox -s***
- <http://packages.ubuntu.com/> - search available Ubuntu packages

Software repositories

- to facilitate software installation and upgrading there will be repositories configured in Ubuntu systems
- in files ***/etc/apt/sources.list*** and ***/etc/apt/sources.list.d/*.list*** are written repository addresses (real servers), having a backup of main repository file would be useful
sudo cp /etc/apt/sources.list /etc/apt/sources.list.backup
- adding repositories
 - uncommenting lines in */etc/apt/sources.list*
 - using command:
 - ***sudo add-apt-repository "deb http://us.archive.ubuntu.com/ubuntu/ xenial universe multiverse"*** # depending on your location, you should replace 'us.' by another country code, referring to a mirror server in your region. Check *sources.list* to see what is used! GPG-keys for official repos are already imported.
 - 3rd party repositories (ensure its sustainability)
 - ***sudo add-apt-repository -y ppa:libreoffice/ppa*** # will automatically import GPG-key
 - <https://launchpad.net/ubuntu/+ppas> - search from PPAs (*Personal Package Archive*)
- before renewing the system there should the software database should be updated after adding any new repository
 - ***sudo apt-get update***

Removing software

- Ubuntu and Debian based systems the software removal can be done by using the following commands
 - ***sudo apt-get remove package***
 - ***sudo apt-get purge package***
(complete removal – including settings)
- also depending packages will be removed
- therefore the parameter -s would be useful (simulate removal) and when the result is fine – then run the actual removal (without -s parameter)

Updating software

- update software list
 - ***sudo apt-get update***
 - ***sudo apt update***
- update existing software
 - ***sudo apt-get upgrade***
 - ***sudo apt upgrade***
- update existing software and intelligently handles changing dependencies with new versions of packages
 - ***sudo apt-get dist-upgrade***
 - ***sudo apt full-upgrade***
- Renew Linux [distributsion](#)
 - ***sudo do-release-upgrade***
 - <http://www.tecmint.com/upgrade-ubuntu-14-04-to-16-04/>

search information about package

- If you are in a situation where you can not remember the specific name of the software, you should use search software packages from the list and description of
- ***sudo apt-cache search description***
 - e.g. ***sudo apt-cache search monitoring***
- ***sudo aptitude search description***
- ***sudo apt search description***

View information of package

- To find out which version of the software is installed, use one of the commands
 - ***apt-cache policy***
 - ***apt-cache showpkg package-name***
 - ***apt-cache show package-name***
- to view dependencies:
 - ***apt-cache showpkg package-name***
 - ***apt-cache depends package-name***
 - ***apt-cache rdepends package-name***
(reverse dependencies)

Software manual installation

- If the appropriate software repository is not available, but the manufacturer's website you will find a deb pack, you can install software using dpkg
 - ***sudo dpkg -i package***
- to solve potential dependency problems:
 - ***sudo apt-get -f install***
- dependency problems are solved automatically by the installer **GDebi** (CLI: *sudo gdebi*, GUI: *gksu gdebi-gtk*)
 - GUI allows to install .deb packages analogous to the MS Windows help (double) clicking and installing
 - GDebi says, among other things, if there are dependencies that can not be solved or .deb package is a newer version already installed

Software installation has been interrupted...

- in case of interrupted software installation
 - install all dependencies
 - **`sudo apt-get -f install`** (also **`sudo apt-get install -f`** works)
 - **`sudo apt -f install`** (also **`sudo apt install -f`** works)
 - configure all installed packages
 - **`sudo dpkg --configure -a`**
 - if there is nothing to do – no harmful thing happens
- having package **dkms** (*Dynamic Kernel Module Support*) installed would be useful for kernel modules (*aka* drivers), especially when there are strongly on specific kernel version dependent modules like proprietary graphics driver, virtualization modules, etc.
 - `sudo apt update && sudo apt install dkms && sudo apt clean`

Management

- downloaded and installed packages remain into cache (*/var/cache/apt/archives/*). To empty APT cache:
 - ***sudo apt-get clean***
 - ***sudo apt clean***
 - remove packages that are not needed anymore
 - ***sudo apt-get autoremove***
 - ***sudo apt purge --auto-remove*** (complete removal)
 - when APT database is busy
 - *E: Could not get lock /var/lib/dpkg/lock - open (11 Resource temporarily unavailable)*
 - *E: Unable to lock the administration directory (/var/lib/dpkg/) is another process using it?*
 - remove your */var/lib/dpkg/lock* file and force package reconfiguration:
 - `sudo rm /var/lib/dpkg/lock`
 - `sudo dpkg --configure -a`
- you may also need to delete:
`sudo rm /var/lib/apt/lists/lock`
`sudo rm /var/cache/apt/archives/lock`

Repositories management

- when there is a message by updating software:
 - `W: GPG error: http://mirrors.dotsrc.org xenial-getdeb Release: The following signatures couldn't be verified because the public key is not available: NO_PUBKEY A8A515F046D7E7CF`
- then one solution:
 - `sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys A8A515F046D7E7CF`
 - `sudo apt update (sudo apt-get update)`
- there are some packages that will add repository when first time to install .deb package: Google Chrome, Google Hangout plugin, MegaSync cloud storage and communication suite, Vivaldi browser, etc.
 - `deb [arch=amd64] http://dl.google.com/linux/chrome/deb/stable main # Google Chrome is currently available only for 64-bit systems`

Kernel management

- `uname -r` shows currently running kernel
- shows installed kernel(s)
 - `dpkg --get-selections | grep linux-image`
 - `dpkg --get-selections | grep linux-headers`

or

- `dpkg -l | grep linux-image`
- `dpkg -l | grep linux-headers`
- usually with software updates also new version of kernel will be installed
- if you want to install LTS (Long Term Support) kernel:
 - `sudo apt search linux-image-generic-lts`
 - `sudo apt search linux-headers-generic-lts`
- Ubuntu codenames: <https://wiki.ubuntu.com/Releases> and <https://wiki.ubuntu.com/DevelopmentCodeNames>
 - `lsb_release -cd`

Kernel management

- installing Ubuntu 16.04 LTS kernel and rebooting
 - `sudo apt update && sudo apt install linux-image-generic-lts-xenial linux-headers-generic-lts-xenial linux-image-generic linux-headers-generic && sudo apt clean && sudo update-grub && sudo reboot`
- removing old kernels
 - using `purge-old-kernels` bundled with `byobu`
 - `sudo purge-old-kernels`
 - `sudo purge-old-kernels --keep 1 -qy` (only the latest)
 - `man purge-old-kernels`
 - installing latest `byobu` (CLI-based window manager, sysadmin's swiss army knife - has GNU Screen built-in transparently to keep sessions running when Internet connection has been interrupted)
 - `sudo add-apt-repository ppa:byobu/ppa && sudo apt-get update && sudo apt install byobu && sudo apt clean`
 - <https://launchpad.net/~byobu/+archive/ubuntu/ppa>

Kernel management

- removing manually old kernels (completely)
 - query installed kernels using CLI:
 - `dpkg-query -f 'linux-image*' | grep '^ii'`
 - `dpkg-query -f 'linux-header*' | grep '^ii'`

or

- `sudo dpkg --get-selections | grep linux-image`
- `sudo dpkg --get-selections | grep linux-header`
- then mark them for complete removal:
 - `sudo apt purge package1 package2 ... package n`
- query installed kernels using GUI (e.g. Synaptic: `sudo apt update && sudo apt-get -y install synaptic && sudo apt clean`) – search the following packages and remove unused packages completely (*SHIFT+Delete*) by selecting them (hold CTRL or SHIFT while clicking on packages):
 - `linux-image`
 - `linux-header`

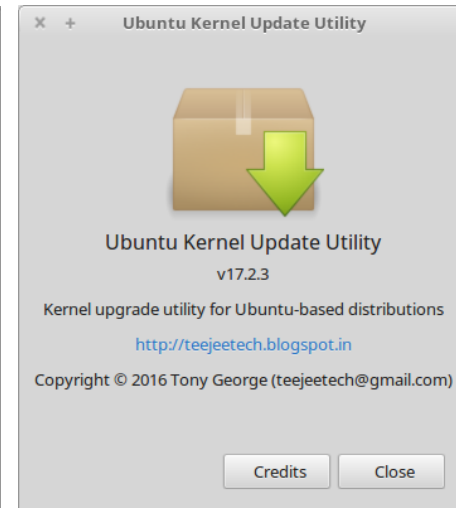
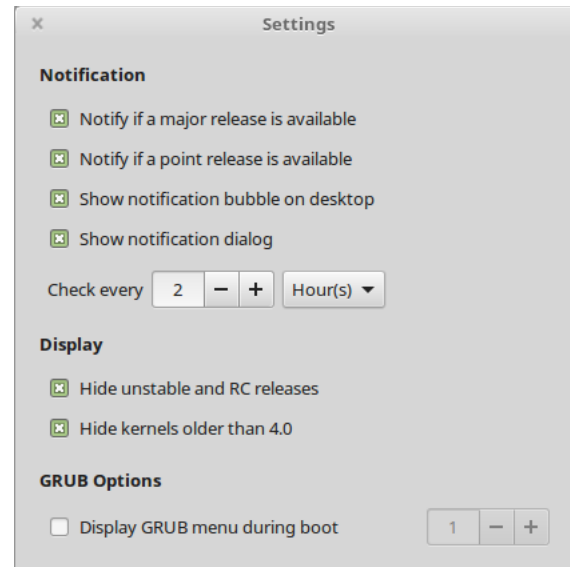
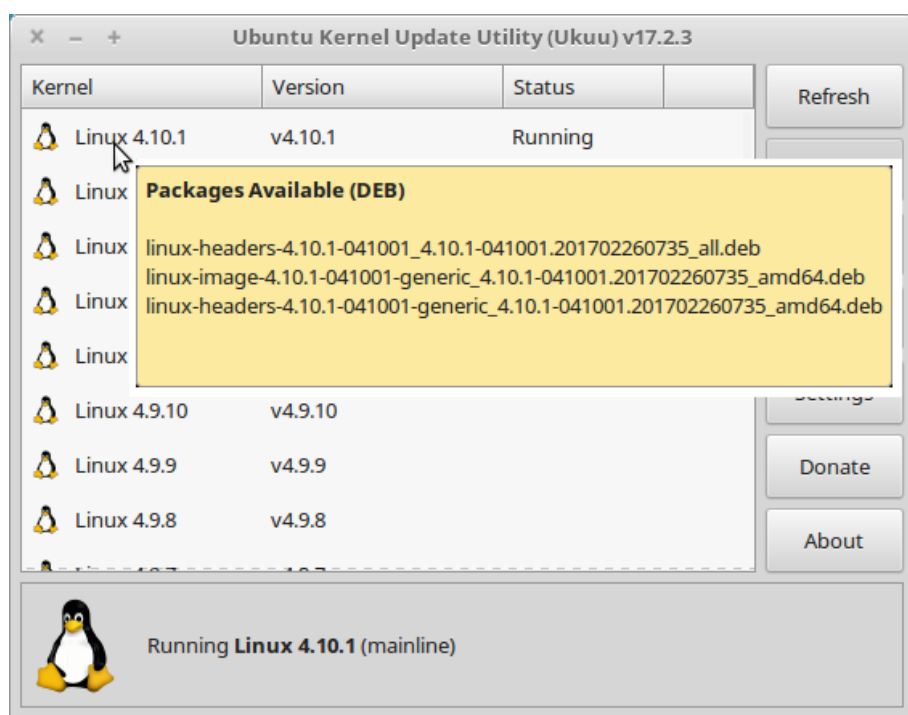
Kernel management

- installing very latest stable kernel via CLI
 - check the latest stable from <https://www.kernel.org/>
 - check .deb packages from <http://kernel.ubuntu.com/~kernel-ppa/mainline/> - take stable releases, *rc* is *Release Candidate*
 - check machine architecture: `arch` OR `uname -m`
 - in case of 32-bit system, download:
 - `linux-headers-VERSION_VERSION.TIME_all.deb`
 - `linux-headers-VERSION-generic_VERSION.TIME_i386.deb`
 - `linux-image-VERSION-generic_VERSION.TIME_i386.deb`
 - if available, also `linux-image-extra-VERSION_i386.deb` can be taken
 - in case of 64-bit system, download:
 - `linux-headers-VERSION_VERSION.TIME_all.deb`
 - `linux-headers-VERSION-generic_VERSION.TIME_amd64.deb`
 - `linux-image-VERSION-generic_VERSION.TIME_amd64.deb`
 - if available, also `linux-image-extra-VERSION_amd64.deb` can be taken

download all .deb packages into one folder and install them by running the following command in that folder:
sudo dpkg -i *.deb
 then update GRUB:
sudo update-grub
 Then reboot machine to start using new kernel
sudo reboot

Kernel management

- installing very latest stable kernel via GUI (on desktop)
 - using Ukuu, the GUI to install latest stable kernel
 - `sudo apt-add-repository -y ppa:teejee2008/ppa && sudo apt-get update && sudo apt-get -y install ukuu && sudo apt-get clean`
 - launch the GUI (`gksu ukuu-gtk`), which is pretty straightforward: refresh repository and install the latest kernel and remove old ones, program detects appropriate architecture



Ubuntu Kernel Update Utility (Ukuu)

- Ukuu notification:

```
~/.config/autostart/ukuu.desktop
```

```
[Desktop Entry]
```

```
Type=Application
```

```
Exec=sh "/home/user/.config/ukuu-notify.sh"
```

```
Hidden=false
```

```
NoDisplay=false
```

```
X-GNOME-Autostart-enabled=true
```

```
Name[en_IN]=Ukuu Notification
```

```
Name=Ukuu Notification
```

```
Comment[en_IN]=Ukuu Notification
```

```
Comment=Ukuu Notification
```

```
~/.config/ukuu-notify.sh
sleep 300s
while true
do
    ukuu --notify ;
sleep 2h
done
```

```
~/.config/ukuu.json
{
    "notify_major" : "true",
    "notify_minor" : "true",
    "notify_bubble" : "true",
    "notify_dialog" : "true",
    "hide_unstable" : "true",
    "hide_older" : "true",
    "notify_interval_unit" : "0",
    "notify_interval_value" : "2",
    "show_grub_menu" : "false",
    "grub_timeout" : "0"
}
```


Automatic updates

- some servers do not have SLA ([Service Level Agreement](#) - service working time, performance), OLA ([Operational-Level Agreement](#), maintenance etc) agreement and installing security updates automatically in such cases might bring more damages than benefit
- mission-critical server updates must be tested beforehand
- sometimes the risk can be with automatic updates and many systems could be configured so (update automatically)
- e.g. Ubuntu server and many other systems will offer to turn on automatic update already during system installation

Automatic updates (2)

- When during installation process this has not been set, it can be configured later:
 - ***sudo apt-get install unattended-upgrades***
 - ***man unattended-upgrade***
- In file ***/etc/apt/apt.conf.d/50unattended-upgrades*** there can be configured automatic updates by removing // from appropriate lines:

// Automatically upgrade packages from these (origin:archive) pairs

Unattended-Upgrade::Allowed-Origins {

"\${distro_id}:\${distro_codename}-security";

// "\${distro_id}:\${distro_codename}-updates";

// "\${distro_id}:\${distro_codename}-proposed";

// "\${distro_id}:\${distro_codename}-backports";

};

Automatic updates (3)

- In file */etc/apt/apt.conf.d/10periodic* (also *20auto-upgrades* in desktop version) there are described updating time and frequency in Ubuntu systems

```
APT::Periodic::Update-Package-Lists "1";
```

```
APT::Periodic::Download-Upgradeable-Packages "1";
```

```
APT::Periodic::AutocleanInterval "7";
```

```
APT::Periodic::Unattended-Upgrade "1";
```

<https://help.ubuntu.com/lts/serverguide/automatic-updates.html>

<http://askubuntu.com/questions/172524/how-can-i-check-if-automatic-updates-are-enabled>

Software installing from source code

- Installing from source code can be achieved in many ways. Whenever possible, installing from package would be suggested. Try to find a PPA or even create own repository.
- Often fits the following pattern:
 - Download software and unpack into certain directory
 - Read the README and INSTALL files within a folder or their analogues
 - Run the commands in the same directory (as the README / INSTALL does not contend otherwise):
 - `./configure`
 - `make`
 - `sudo make install`
 - `make clean` (if you want to start over)

Installation cloning

- Saving installed packages list
 - 1.variant
 - `dpkg --get-selections > installed-packages.txt`
 - 2.variant
 - `apt list --installed > installed-packages.txt`
- Installing packages from saved packages list
 - 1. variant
 - `sudo dpkg --set-selections < installed-packages.txt`
 - 2.variant
 - `sudo dpkg --clear-selections && sudo dpkg --set-selections < installed-packages.txt && sudo apt-get -u dselect-upgrade`
- using appropriate software, e.g. Aptik (desktop) -
<http://www.tecmint.com/aptik-a-tool-to-backuprestore-your-favourite-ppas-and-apps-in-ubuntu/> also
<http://www.makeuseof.com/tag/10-easy-ways-restore-linux-system/> - there are also CLI tools

References

- Ubuntu server automatic updates
 - <https://help.ubuntu.com/lts/serverguide/automatic-updates.html>
 - <https://help.ubuntu.com/community/AutomaticSecurityUpdates>
- Ubuntu server upgrading
 - <https://help.ubuntu.com/lts/serverguide/installing-upgrading.html>
 - <https://help.ubuntu.com/community/Upgrades>
- Software management in Linux
 - http://www.linuxtopia.org/online_books/linux_administrators_security_guide/14_Linux_Software_Management.html
 - <https://help.ubuntu.com/community/InstallingSoftware> , please see software <https://apps.ubuntu.com/>
- Corresponding package manager for MS Windows like in Linux:
 - <http://www.howtogeek.com/141783/how-to-bring-linux-style-apt-get-installations-to-windows-with-chocolatey/>
 - <https://winstall.com/>
- *sudo* alternative for MS Windows:
 - <http://superuser.com/questions/42537/is-there-any-sudo-command-for-windows>
 - <http://helpdeskgeek.com/free-tools-review/5-windows-alternatives-linux-sudo-command/>

References

- creating, managing own APT repository
 - https://debian-administration.org/article/286/Setting_up_your_own_APT_repository_with_upload_support
 - <https://wiki.debian.org/RepositoryFormat>
 - <https://wiki.debian.org/HowToSetupADebianRepository>
 - <https://help.ubuntu.com/community/Repositories/Personal>
 - <https://help.ubuntu.com/community/CreateAuthenticatedRepository>
 - <https://wiki.debian.org/SettingUpSignedAptRepositoryWithReprepro>
 - <http://unix.stackexchange.com/questions/87130/how-to-quickly-create-a-local-apt-repository-for-random-packages-using-a-debian>
 - <https://www.aptly.info> - is a swiss army knife for Debian repository management
- kernel news
 - <https://www.kernel.org/>
 - <https://lwn.net/Kernel/>
 - <https://lkml.org/> - mailing list
 - http://www.phoronix.com/scan.php?page=news_topic&q=Linux+Kernel
 - <https://www.reddit.com/r/kernel/>
 - <https://kernelnewbies.org/>

Questions?

Thank you for your attention!

