Tishk International University
Science Faculty
IT Department

Open Source OS (Linux)

Lecture 05: Process and Package Management

4th Grade - Fall Semester

Instructor: Alaa Ghazi

Lecture 5
Process and Package Management

Topics

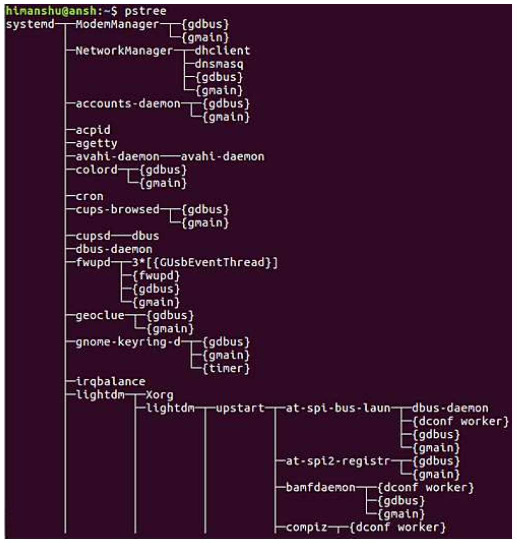
1 Basics of Processes

- **A Process** is an instance of a running program, and there can be multiple instances of the same application
- Linux is a multi-tasking OS, can run multiple processes simultaneously and each process may have multiple threads, where a thread is just another process (of special nature)
- Each user starting a process becomes its owner
- The process owner does not have to be the same as the owner of the binary file for the process
- Each process have an owner, some processes started by the system can be owned by the root user
- The process owner has privileges on his process like (kill, pause, resume)
- The 'root' user have super powers on all system processes
- The process inherits its user privileges when trying to access resources
- (for example when a process tries to write in a file)

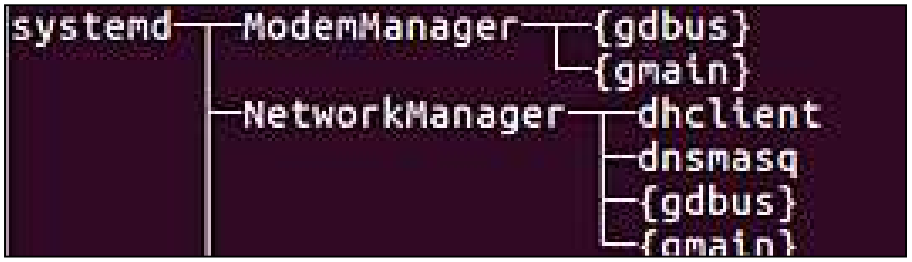
2 Parent & Child Processes

- Each process that creates another becomes the parent, and the new process becomes the child process
- ***systemd*** process is started at system boot and it is the grand parent of all processes in the whole system
- Process ID (PID) is a unique number to identify the process
- Each process will maintain its **PID** and the **PID** of its parent ***PPID***
- The **systemd** process has **PID** = 1 and **PPID** = 0
- **Process Group** is a family of processes (A process, its children, grand-children, …etc) and when a process is created it becomes a member of the process group of its parent
- **pstree** is the command to show the Process tree hierarchy

- ***$ pstree***(Show tree starting at **systemd** process)
- ***$ pstree -p*** (to show PIDs of all processes)
- ***$ pstree 1025*** (Show tree starting at process with PID = 1025)



```
himanshu@ansh:~$ pstree
systemd─┬─ModemManager─┬─{gdbus}
        │              └─{gmain}
        ├─NetworkManager─┬─dhclient
        │                ├─dnsmasq
        │                ├─{gdbus}
        │                └─{gmain}
        ├─accounts-daemon─┬─{gdbus}
        │                 └─{gmain}
        ├─acpid
        ├─agetty
        ├─avahi-daemon───avahi-daemon
        ├─colord─┬─{gdbus}
        │        └─{gmain}
        ├─cron
        ├─cups-browsed─┬─{gdbus}
        │              └─{gmain}
        ├─cupsd───dbus
        ├─dbus-daemon
        ├─fwupd─┬─3*[{GUsbEventThread}]
        │       ├─{fwupd}
        │       ├─{gdbus}
        │       └─{gmain}
        ├─geoclue─┬─{gdbus}
        │         └─{gmain}
        ├─gnome-keyring-d─┬─{gdbus}
        │                 ├─{gmain}
        │                 └─{timer}
        ├─irqbalance
        ├─lightdm─┬─Xorg
        │         ├─lightdm─┬─upstart─┬─at-spi-bus-laun─┬─dbus-daemon
        │         │         │         │                 ├─{dconf worker}
        │         │         │         │                 ├─{gdbus}
        │         │         │         │                 └─{gmain}
        │         │         │         ├─at-spi2-registr─┬─{gdbus}
        │         │         │         │                 └─{gmain}
        │         │         │         ├─bamfdaemon─┬─{dconf worker}
        │         │         │         │            ├─{gdbus}
        │         │         │         │            └─{gmain}
        │         │         │         ├─compiz─┬─{dconf worker}
```



```
systemd─┬─ModemManager─┬─{gdbus}
        │              └─{gmain}
        ├─NetworkManager─┬─dhclient
        │                ├─dnsmasq
        │                ├─{gdbus}
        │                └─{gmain}
```

3 Process Types

- Processes can be classified into one of the following,
- **Interactive Process**: is started by a user within a terminal and It is attached to its terminal, and will be killed if its terminal is closed
- **Automatic Process(Batch Process)** is not started directly by the user, instead, the user schedule it for a later start and it is not attached to terminal

- **Daemon Process**: is a process that runs continuously in the background toperform a task, or waiting for services to be requested from it and they normally start at system startup)

3 The Job Control

- **Job** is the execution of a command in a terminal and it can be a single process or multiple connected processes
- Jobs can run in the below modes:
- **Foreground Job**: All input and output of the terminal is exclusively for this job
- To Start a job in the foreground just type the command like

```
firefox
```

- **Background Job**: the Input and output do not utilize the terminal, and multiple Jobs can be in the background for the same terminal
- To start a job in the background use & sign at the end

```
firefox &
```

- To pause the foreground Job use ***Ctrl-z***
- To resume the paused Job in the foreground use

```
fg
```

- To Interrupt and stop a foreground Job use ***Ctrl-c***

4 Process Attributes

- **The process ID or PID**: a unique identification number used to refer to the process.
- **The parent process ID or PPID**: the number of the process (PID) that started this process.
- **Nice number**: the degree of friendliness of this process toward other processes
- **Terminal or TTY**: terminal to which the process is connected.
- **Real User ID (RUID)**: the user who started the process
- **Effective User ID (EUID)**: the user whose privileges will be inherited by the process when accessing resources
- **Real Group ID (*RGID*)**: The primary group of the user who started the process
- **Effective Group ID (*EGID*):** The Group ID whose privileges are inherited by the process when accessing system resources
- NOTE: By default RUID and EUID has the same value also **RGID** and **EGID.**

5 ps Command: Process Status

- **$ ps <options>**

-e Everything, all processes

-f Full format listing.

-u username Display username’s processes.

-p pid Display information for PID

- **Examples**

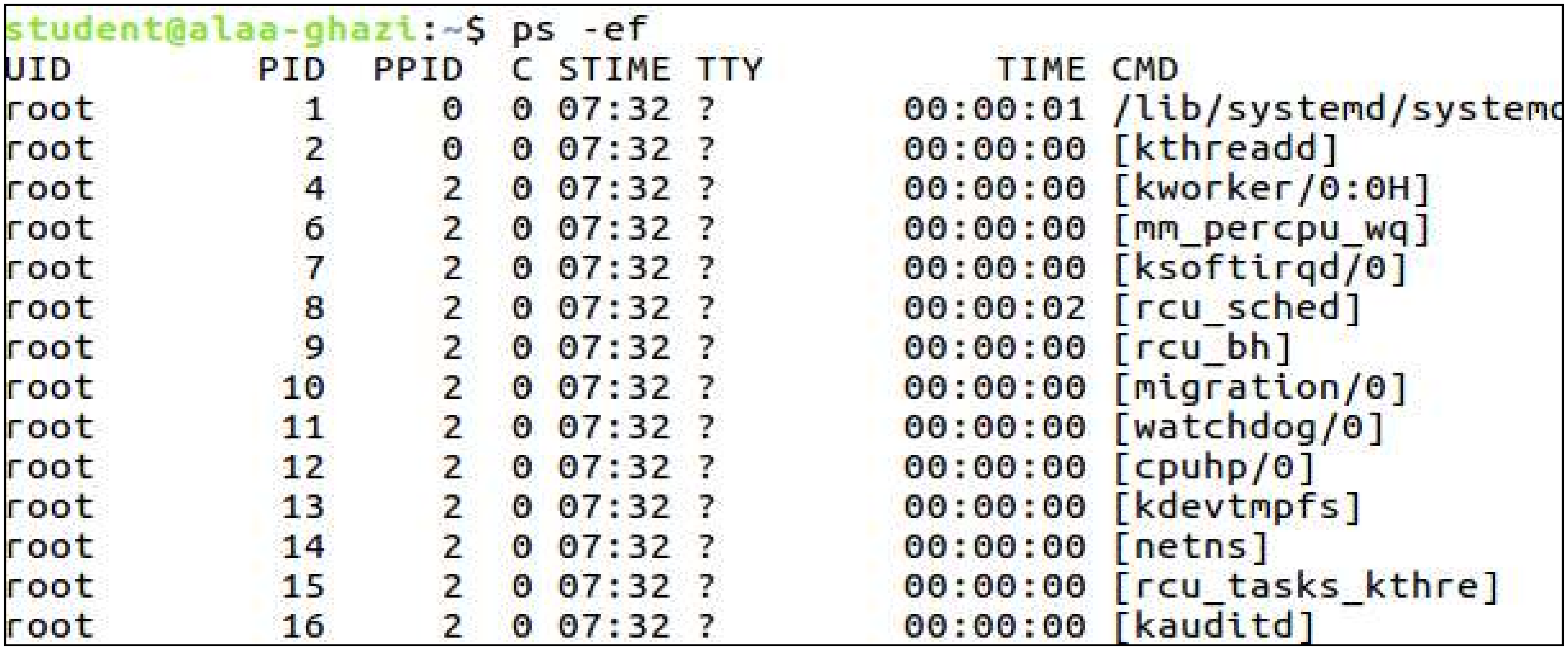
ps -e Display all processes.

ps -ef Display all processes, full.

ps -eH Display a process tree.

ps -e --forest Display a process tree.

ps -u username Display user’s processes.



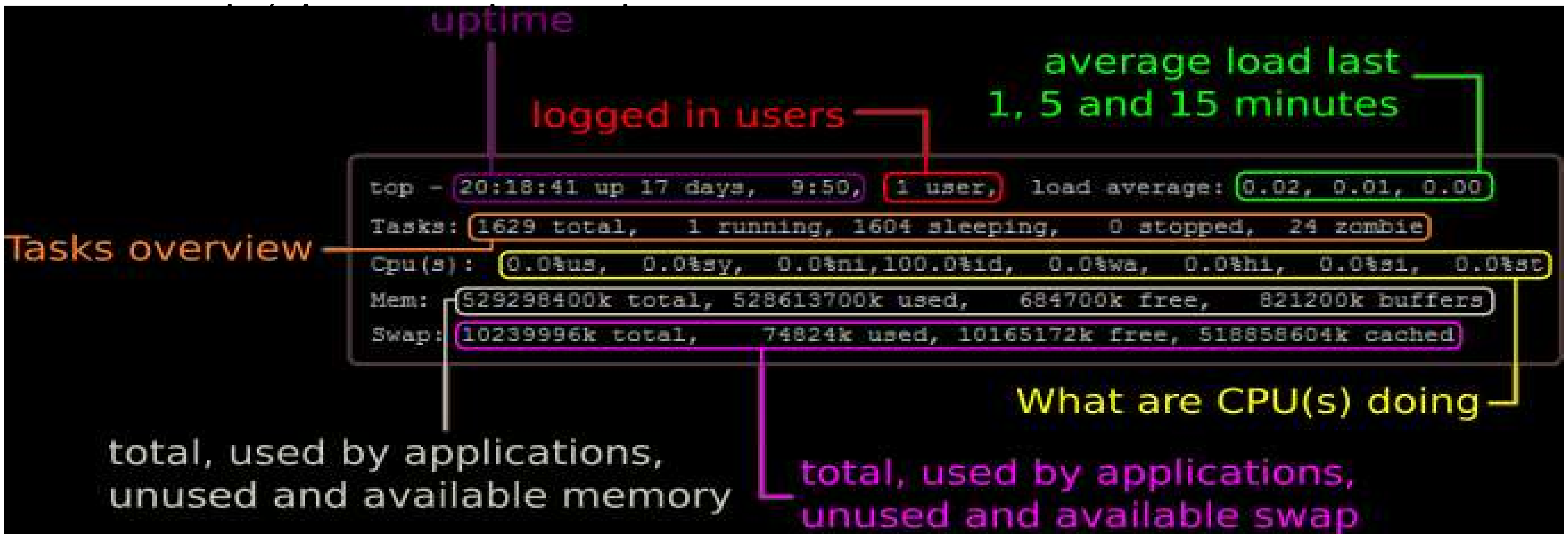
```
student@alaa-ghazi:~$ ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 07:32 ?        00:00:01 /lib/systemd/systemd
root           2       0  0 07:32 ?        00:00:00 [kthreadd]
root           4       2  0 07:32 ?        00:00:00 [kworker/0:0H]
root           6       2  0 07:32 ?        00:00:00 [mm_percpu_wq]
root           7       2  0 07:32 ?        00:00:00 [ksoftirqd/0]
root           8       2  0 07:32 ?        00:00:02 [rcu_sched]
root           9       2  0 07:32 ?        00:00:00 [rcu_bh]
root          10       2  0 07:32 ?        00:00:00 [migration/0]
root          11       2  0 07:32 ?        00:00:00 [watchdog/0]
root          12       2  0 07:32 ?        00:00:00 [cpuhp/0]
root          13       2  0 07:32 ?        00:00:00 [kdevtmpfs]
root          14       2  0 07:32 ?        00:00:00 [netns]
root          15       2  0 07:32 ?        00:00:00 [rcu_tasks_kthre]
root          16       2  0 07:32 ?        00:00:00 [kauditd]
```

6 top command

•**top Command:** Displays a dynamic view of the resource usage of system processes .

$ top

- While the "***top***" tool is running,
  - Push 'M' to sort by memory usage
  - Push 'P' to sort by CPU processing usage
  - Push 'T' to sort by Time
  - Push 'k <pid>' to kill process by its pid
  - Push 'h' for getting a help page for all options
  - Push 'H' to enable/disable showing threads separately


uptime
average load last 1, 5 and 15 minutes
logged in users
Tasks overview
What are CPU(s) doing
total, used by applications, unused and available memory
total, used by applications, unused and available swap

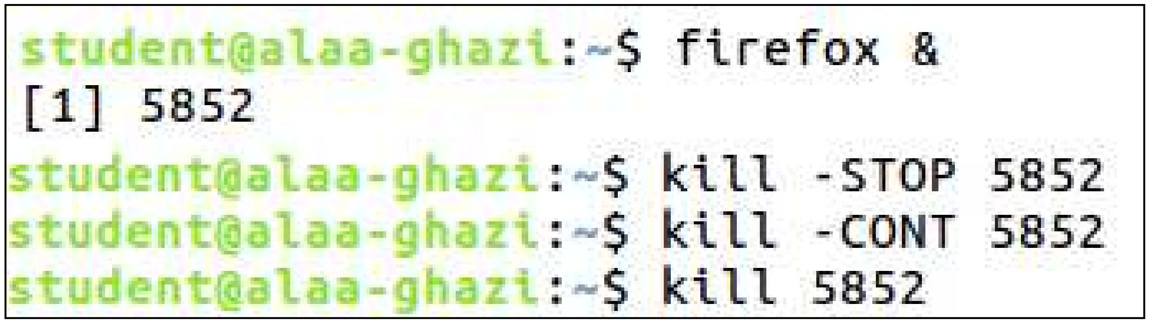
```
top - 20:18:41 up 17 days,  9:50,  1 user,  load average: 0.02, 0.01, 0.00
Tasks: 1629 total,   1 running, 1604 sleeping,   0 stopped,  24 zombie
Cpu(s):  0.0%us,  0.0%sy,  0.0%ni,100.0%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:  529298400k total, 528613700k used,   684700k free,   821200k buffers
Swap: 10239996k total,    74824k used, 10165172k free, 518858604k cached
```

7 Inter-Process Communication

- **Inter-Process Communication IPC**: Allow processes to exchange information.
- IPC mechanisms:

7.1 Signals: one word exchanged between processes and Kernel uses them to notify processes when certain events in response to interrupts and exceptions

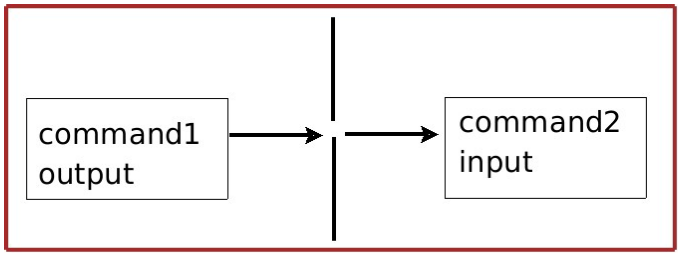
- A process or thread can block a signal
- Example of a command that uses signal is kill
- ***kill*** command is a built-in command which sends a signal to a process to terminates the process.
- kill sends TERM signal by default but it can be used to send ay other signal, like in examples below: **How to suspend firefox process, resume it and then terminate it using process PID.**



```
student@alaa-ghazi:~$ firefox &
[1] 5852
student@alaa-ghazi:~$ kill -STOP 5852
student@alaa-ghazi:~$ kill -CONT 5852
student@alaa-ghazi:~$ kill 5852
```

Inter-Process Communication

7.2 Pipes: it is a mechanism in which Producer process writes data to the pipe, after which the consumer process reads data from the pipe in first-in-first-out queue.



- **Piping Examples**

ls –l | more

ps –e | grep root

7.3 Sockets: Allows pairs of processes on the same system or different systems to exchange data by establishing direct bidirectional communication channels

- Socket Types:
  - **Stream sockets**: Implement the traditional client/server model and they use TCP for reliable communication
  - **Datagram sockets**: Faster, but less reliable communication and use UDP packets

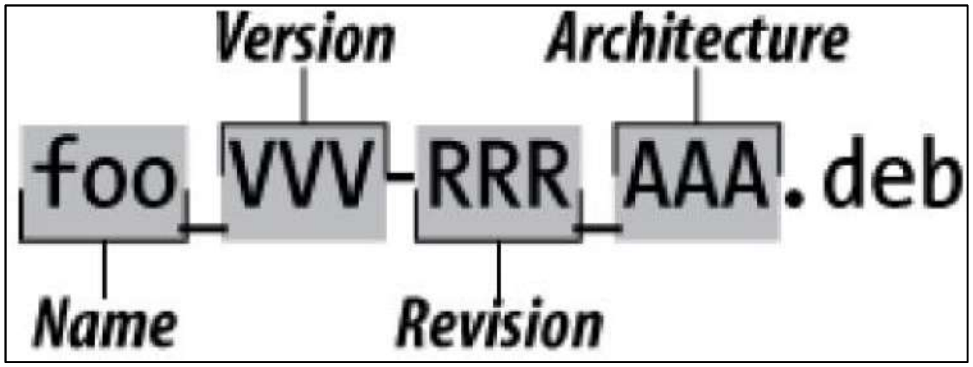
8 Package Management Overview

- Package Management is A way to distribute software and configuration
- Eg.
  - .tar.gz or tgz (Slackware)
  - .rpm (Red Hat, Fedora, SUSE, …)
  - .deb (Debian, Ubuntu)
- Meta-package manager
  - Locate packages on the Internet, download, install and analyze inter-package dependencies. eg.
    - yum (rpm)
    - apt-get (deb and rpm)

9 Software Packages

- **Build from the source code:** Need to download the software and compile it to generate the binaries/libraries
- **Install a pre-prepared Package:** The software may be pre-prepared in a package that is ready for installation and it is Only applicable to official releases
  - Packages depend on the used Linux distribution
    - Debian based distributions (like Ubuntu) they come in ***.deb***
    - Red Hat based distributions (like Fedora) they come in ***.rpm***
- **A Software Package** is an archive of files that contains, The binary files to be installed (previously built) , configuration files, Meta data about the package (including a list of dependencies) , and installation scripts
- Packages are available,
  - Individually on internal sites (a package file with extension ***.deb*** or ***.rpm***)
  - Among a group within common repositories (collection of packages)
- Tools and package format is dependent on the Linux distribution (we will focus on Ubuntu based distributions)

10 Package File Name Format



- The package name normally contains words separated by hyphens
- The package version is normally composed of 3 numbers separated by dots, in the format **major.minor.patch**
- The Architecture is normally to state what kind of processor this
- package is targeting
- Examples are:

  gedit-common_3.10.4-0ubuntu4_i386.deb
  gnome-user-guide_3.8.2-1_all.deb
  Libfile-copy-recursive-perl_0.38-1_all.deb
  2048-qt_0.1.6-1_amd64.deb

11 dpkg Command

- To install .deb file, we use the tool ***dpkg*** which can be used as below

dpkg -i <package file>

- To remove the packge we use

dpkg -r <package name>

- If we have a package named "package name",
- To list all installed packages

dpkg -l

- To show all files inside a package

dpkg -L <package name>

- To determine if a package is installed or not

dpkg --status <package name>

- To find out which package installed a certain file

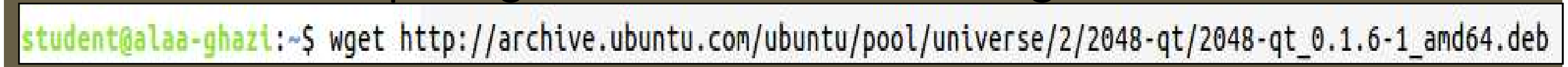
dpkg --search <file name>

- To fix dpkg problems run

dpkg --configure –a

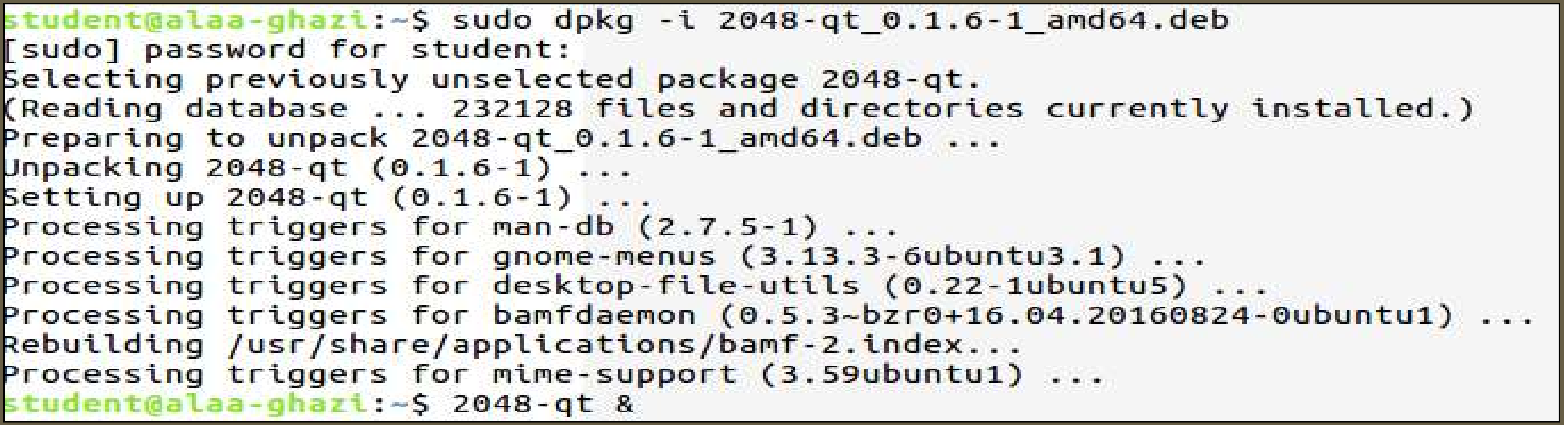
Example: using dpkg command

- To download the package in terminal we use the **wget <ink>**



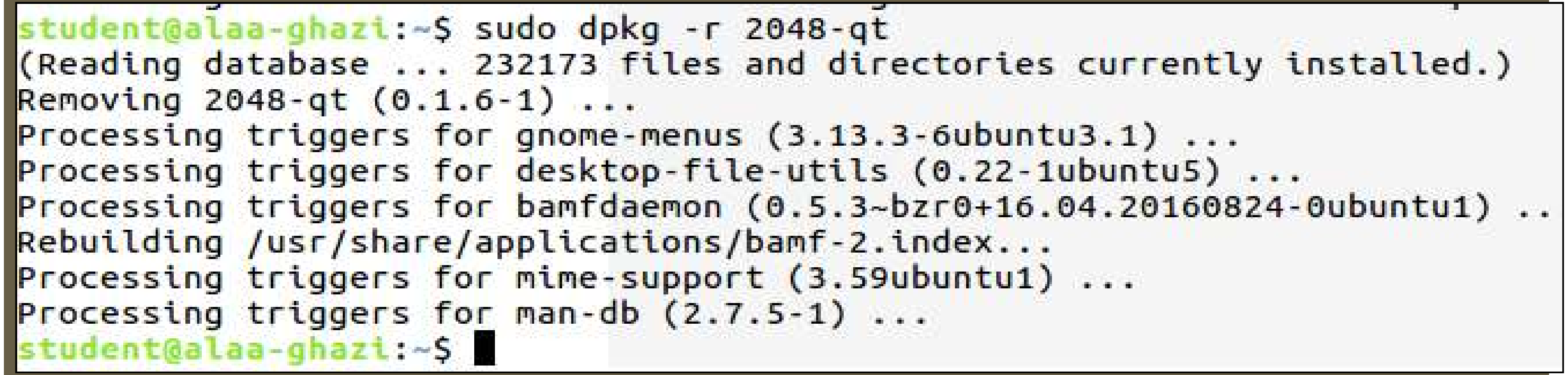
```
student@alaa-ghazi:~$ wget http://archive.ubuntu.com/ubuntu/pool/universe/2/2048-qt/2048-qt_0.1.6-1_amd64.deb
```

- After downloading the package file, we can install it as follows, and then run it



```
student@alaa-ghazi:~$ sudo dpkg -i 2048-qt_0.1.6-1_amd64.deb
[sudo] password for student:
Selecting previously unselected package 2048-qt.
(Reading database ... 232128 files and directories currently installed.)
Preparing to unpack 2048-qt_0.1.6-1_amd64.deb ...
Unpacking 2048-qt (0.1.6-1) ...
Setting up 2048-qt (0.1.6-1) ...
Processing triggers for man-db (2.7.5-1) ...
Processing triggers for gnome-menus (3.13.3-6ubuntu3.1) ...
Processing triggers for desktop-file-utils (0.22-1ubuntu5) ...
Processing triggers for bamfdaemon (0.5.3~bzr0+16.04.20160824-0ubuntu1) ...
Rebuilding /usr/share/applications/bamf-2.index...
Processing triggers for mime-support (3.59ubuntu1) ...
student@alaa-ghazi:~$ 2048-qt &
```

- If we later decide to un-install the package, we do,



```
student@alaa-ghazi:~$ sudo dpkg -r 2048-qt
(Reading database ... 232173 files and directories currently installed.)
Removing 2048-qt (0.1.6-1) ...
Processing triggers for gnome-menus (3.13.3-6ubuntu3.1) ...
Processing triggers for desktop-file-utils (0.22-1ubuntu5) ...
Processing triggers for bamfdaemon (0.5.3~bzr0+16.04.20160824-0ubuntu1) ..
Rebuilding /usr/share/applications/bamf-2.index...
Processing triggers for mime-support (3.59ubuntu1) ...
Processing triggers for man-db (2.7.5-1) ...
student@alaa-ghazi:~$
```

12 Problems with dpkg

- As we have shown, the dpkg tool takes care of installing a package file however, there is a problem,
- A lot of packages have **dependencies,** and the user needs to know the dependencies and perform the required pre-requisites before installing the desired package, which makes the process too complicated.
- We need a high level tool that takes care of dependency resolution
- It should be able to query for the required dependencies
- Then it should perform the needed installations
- Then it installs the desired package
- All of this in a transparent way without user intervention
- This resulted in the tool ***apt***
- **Advanced Packaging Tool "apt"** is a set of high level tools for installing software packages in Debian based Linux distributions

13 Installing Packages using apt command

- To install a certain program or library, all you need is to know the package name that contains it via web search
- Then use the command for installing a package:

sudo apt-get install <package name> -y

- The **apt** tool then performs all the necessary functions
  - It identifies the latest version
  - It identifies any pre-requisites
  - It calculates how much disk space needed
  - It prompts the user to approve the installation
  - It downloads all the needed files from the internet
  - It performs procedure

14 Un-Installing Packages using apt

- To Un-Install a package, and keeping the Configuration files (for future re-installation)

sudo apt-get remove <package name>

- To Un-Install a package, and remove the Configuration files

sudo apt-get purge <package name>

- To remove packages that are no longer needed (they were installed as dependencies but no longer needed)

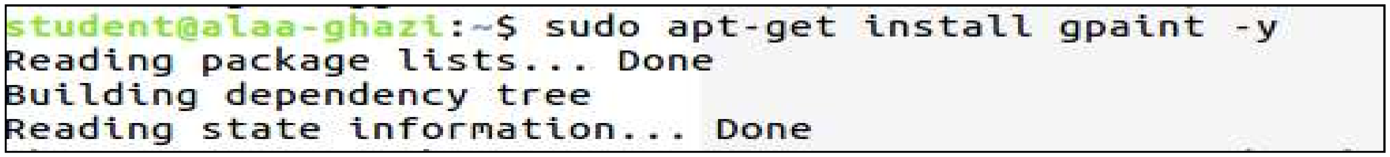
sudo apt-get autoremove

- To fix problems use the command

sudo apt-get -f install

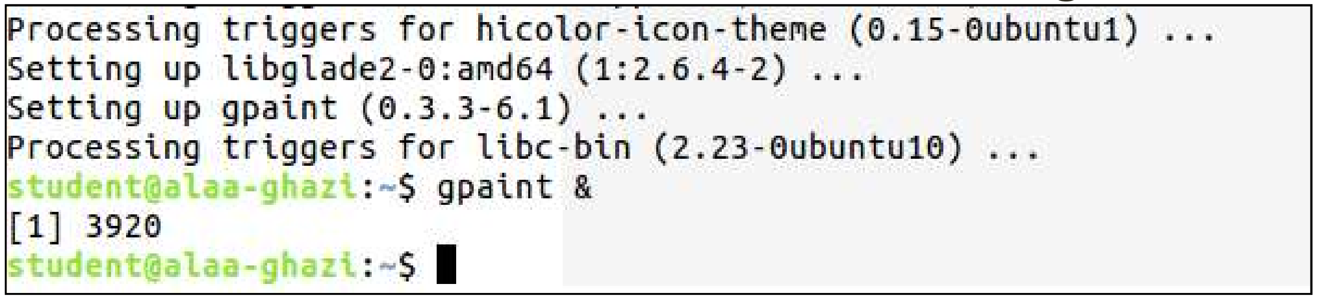
Example: using apt command

- Installing a drawing tool gpaint

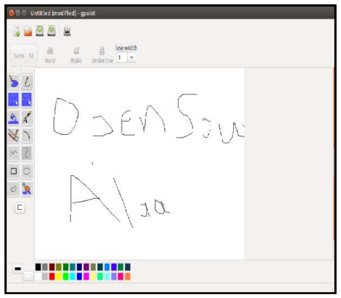


```
student@alaa-ghazi:~$ sudo apt-get install gpaint -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

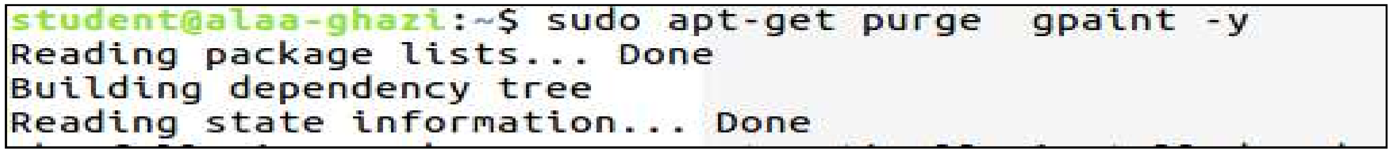
- After installation finishes run the tool in background



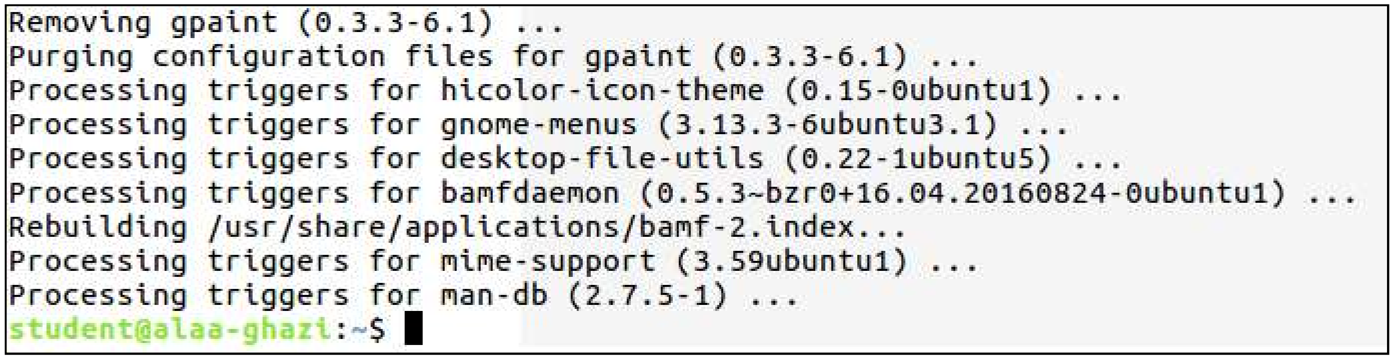
```
Processing triggers for hicolor-icon-theme (0.15-0ubuntu1) ...
Setting up libglade2-0:amd64 (1:2.6.4-2) ...
Setting up gpaint (0.3.3-6.1) ...
Processing triggers for libc-bin (2.23-0ubuntu10) ...
student@alaa-ghazi:~$ gpaint &
[1] 3920
student@alaa-ghazi:~$ █
```



- We can uninstall the package using below:



```
student@alaa-ghazi:~$ sudo apt-get purge  gpaint -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
```



```
Removing gpaint (0.3.3-6.1) ...
Purging configuration files for gpaint (0.3.3-6.1) ...
Processing triggers for hicolor-icon-theme (0.15-0ubuntu1) ...
Processing triggers for gnome-menus (3.13.3-6ubuntu3.1) ...
Processing triggers for desktop-file-utils (0.22-1ubuntu5) ...
Processing triggers for bamfdaemon (0.5.3~bzr0+16.04.20160824-0ubuntu1) ...
Rebuilding /usr/share/applications/bamf-2.index...
Processing triggers for mime-support (3.59ubuntu1) ...
Processing triggers for man-db (2.7.5-1) ...
student@alaa-ghazi:~$ █
```

15 Software Repository

- **Package Repositories** is a collection of packages along with some index file to organize them
- The ***apt*** tool keeps track of which repositories to search for the desired package via a configuration file **/etc/apt/sources.list**
- **/etc/apt/sources.list** contains a list of the URLs for the servers containing the different repositories to search for packages
- User can add/remove repositories by editing this file
- **$ apt-get update**
- This command causes ***apt***, to rebuild its package database
- It goes into the ***/etc/apt/sources.list***
- Queries each repository for the packages it contain
- For each package, get,
  - Latest release number
  - Size
  - Dependency list

LAB 05
Process and Package Management

LAB 05 TEST1: Processes Monitoring

Provide screen shot and comment for each command below:

1) pstree
2) ps –ef
3) top
    - sort by memory usage
    - sort by CPU processing usage
4) In top command find out
    - uptime
    - logged in users
    - total tasks
    - total memory
    - total swap

LAB 05 TEST2: Package Installation using dpkg

1) Take screen shot with comments for each step
2) Open a new terminal
3) Using wget command, download the zoom package from below link

   https://zoom.us/client/5.0.398100.0427/zoom_amd64.deb

3) Try to install the zoom package using dpkg tool
4) After the installation fail identify the dependency
5) Search and Locate the link for the dependency through firefox
6) Try to download and install the dependency packages using dpkg tool
7) Then try to install the zoom package using dpkg tool
8) If failed again repeat the steps till success.

LAB 05 TEST3: Package Installation using apt

1) Open a new terminal
2) Using apt command, install the package xpaint
3) Run xpaint in background mode
4) From the xpaint select canvas> new canvas
5) Draw your name and take screen shot
6) Suspend xpaint and then resume it
5) Kill xpaint in terminal
6) Remove xpaint using apt command
7) Take screen shot with comments for each step