

System Administration



Session 4 CONTENT

- Initialization files
- Installing packages
- Processes
- Listing processes
- Searching for processes
- Signals
- Services
- Logs

Initialization files

When the shell is invoked, there are certain initialization/startup files it reads which help to setup an environment for the shell itself and the system user.

There are two categories of initialization files read by the shell:

- **system-wide startup files** – contain global configurations that apply to all users on the system, and are usually located in the /etc directory.
 - /etc/profiles
 - /etc/bashrc or /etc/bash.bashrc
- **user-specific startup files** – these store configurations that apply to a single user on the system and are normally located in the users home directory as dot files. [They can override the system-wide configurations]
 - .bash_profile
 - .bashrc

Installing packages

A package in linux OS is like an application or a program and there are different ways to install packages.

Each Linux distribution has its own package management system.

yum, rpm

apt, dpkg

```
apt search <application_name>
```

```
dpkg -i <package_name>.deb
```

```
yum install <application_name>
```

```
rpm -i <package_name>.rpm
```

Process

Every program you run creates a process. For example

- Shell
- An application

System starts processes called **daemons** which are processes that run in the background and provide services

Every processes has a **PID**

When a process creates another, the first is the **parent** of the new process. The new process is called the **child process**. Parent waits for child process till it finishes.

All processes are descendants of the first system process, which is systemd

Viewing Processes

Process status command

```
ps option(s)
```

Viewing processes with `top` utility

Searching for Process

Using pgrep command

```
pgrep option(s) pattern
```

Options

- x: exact match
- u uid: processes for a specific user
- l: display the name with pid

Sending a signal

Using kill command

Default signal 15

```
kill [-signal] PIDs
```

Using pkill command

```
pkill [-signal] process_name
```


Foreground & Background

Processes can be launched in the background in order to have the ability to use the terminal while running a long term process

`command &`

You can list the background running/stopped processes

`jobs command`

Controlling services

System starts processes called **daemons** which are processes that run in the background and provide services performing a particular task

You can control the services using

```
systemctl [option] servicename  
service servicename [option]
```

Start/Stop services

Start a service

```
systemctl start service_name  
systemctl start network
```

Stop a service

```
systemctl stop service_name  
systemctl stop network
```

To check status of a service

```
systemctl status service_name  
systemctl status network
```

Enable/Disable services

- To enable auto-start a service at boot

```
systemctl enable service_name  
systemctl enable network
```

- To disable auto-start a service at boot

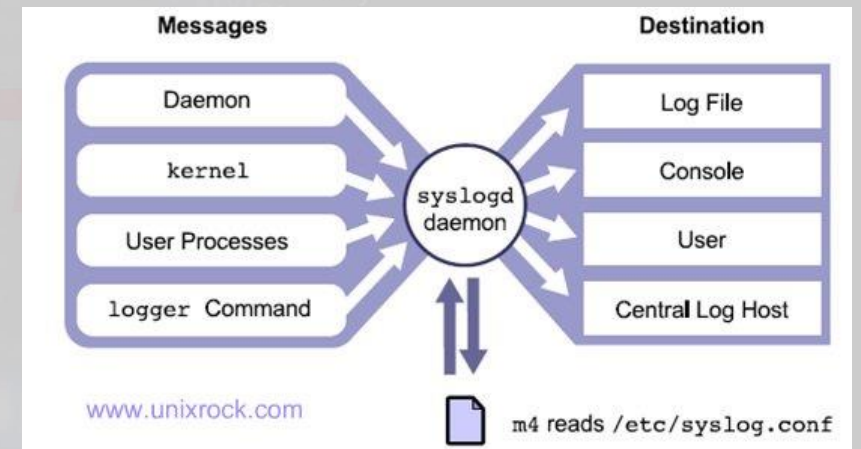
```
systemctl disable service_name  
systemctl disable network
```

Logs

The rsyslog system messaging feature track system activities and events.

System logging:

- | | |
|----------------------|--------------|
| 1- Direct write | (ex: Apache) |
| 2- Through systemctl | (journald) |
| 3- Through rsyslogd | (/var/log) |



You can monitor the designated rsyslog file in real time using the command `tail -f`.

Questions?!

Thank YOU!