RHCEv8 Online Class 29052021 10:00pm
RHCSA-sysadmin
- Monitoring and Managing Linux Processes
- Controlling Services and Daemons
- Managing Networking

Monitoring and Managing Linux Processes

what is process?

A *process* is a running instance of a launched, executable program.
ex:
ls
firefox

Process Type:

2types are available
1-shell process
2-daemon process

1-shell process

commands as run through shell, they are shell processes.
printenv
echo $SHELL
/bin/bash

2-daemon process

-they are system related background process running under the root privilege.
-they will start with OS and stop with OS

how processes born

1-fork system
2-exec

1-fork system

caller(parent) will call to kernel to get permission ->(**call system**), kernel allows and new process (child process) will born in parent's duplicate address space in memory.
firefox
sleep 4000
vim /etc/passwd

2-exec

in this method when child will close, parent(caller) will close too.
exec useradd user1

NOTE:
when new process will have born process-id-pid will generate in 4,5 unique digits and assign on it.\
pid->process-id
ppid->parent process-id

Monitor the Process

-offline ->what happend
ps ->**P**rocess **S**tatus
ps -o ppid,pid,uid,user,tty,stime,time,cmd

PPID	PID	UID	USER	TT	STIME	TIME	CMD
1918	1921	0	root	pts/0	10:20	00:00:00	-bash

tty

-online
top
-to get help press Shift+?
-to exit from help, press Esc key
-to exit press 'q'

htop
nmon
glance
bashtop
atop

yum repolist
yum list glance*
yum install glances.noarch -y
glance
-to exit press 'q'

Install elinks (text-based Web browser) on linux
-get elinks from internet
http://mirror.centos.org/centos/8/PowerTools/x86_64/os/Packages/elinks-0.12-0.58.pre6.el8.x86_64.rpm

wget http://mirror.centos.org/centos/8/PowerTools/x86_64/os/Packages/elinks-0.12-0.58.pre6.el8.x86_64.rpm
rpm -qpi elinks-0.12-0.58.pre6.el8.x86_64.rpm
rpm -qp --scripts elinks-0.12-0.58.pre6.el8.x86_64.rpm
install
rpm -iv elinks-0.12-0.58.pre6.el8.x86_64.rpm
or
yum localinstall elinks-0.12-0.58.pre6.el8.x86_64.rpm -y
elinks

3important shortcuts:
1- Ctrl+c terminates
2- Ctrl+d save/quit
3- Ctrl+z stop

run process at foreground
in fg process, caller won't accessible
sleep 1000
or
elinks
run process at background
in bg process, caller will accessible
sleep 1000 &
[1] 25656

to watch bg process list
jobs
[1]+ Running sleep 1000 &
[1] ->job-id
+ ->freshest process at bg
- ->one process before freshest
Running ->process status
sleep 1000 & ->process name

sleep 10000 &
[2] 25661
jobs
[1]- Running sleep 1000 &
[2]+ Running sleep 10000 &

sleep 100000 &
[3] 25662
jobs
[1] Running sleep 1000 &
[2]- Running sleep 10000 &
[3]+ Running sleep 100000 &

handle process between foreground and background

```
# jobs
[1]   Running              sleep 1000 &
[2]-  Running              sleep 10000 &
[3]+  Running              sleep 100000 &
```

bg to fg

```
# fg %2
sleep 10000
```

fg to bg

```
# Ctrl+z              to stop it
^Z
[2]+  Stopped              sleep 10000
# bg %2
[2]+ sleep 10000 &
# jobs
[1]   Running              sleep 1000 &
[2]-  Running              sleep 10000 &
[3]+  Running              sleep 100000 &
#
```

Monitor the Process continue.

```
 # ps      -a        about all users
           -x        process without terminal
           -f        full
           -u        additional info
           -e        extended info
# ps -aux
# ps -ef
# ps lax
# ps -lax

# ps -aux
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
-user
-pid
-%cpu
-%mem
-vsz virtual set size          ->total memory
-rss resident set size         ->how much using now by process
-tty
   ?                  ->background process
  /dev/pts/0         ->running through tty0
-state
D    uninterruptible sleep (usually IO)
I    Idle kernel thread
R    running or runnable (on run queue)
S    interruptible sleep (waiting for an event to complete)
T    stopped by job control signal
t    stopped by debugger during the tracing
W    paging (not valid since the 2.6.xx kernel)
X    dead (should never be seen)
Z    defunct ("zombie") process, terminated but not reaped by its parent
```

For BSD format:

```
<    high-priority (not nice to other users
N    low-priority (nice to other users)
L    has pages locked into memory (for real-time and custom IO)
s    is a session leader
l    is multi-threaded (using CLONE_THREAD, like NPTL pthreads do)
+    is in the foreground process group

-start              ->what time process started
-time               ->how much time took to start
-command            ->command
```

kill process

what is signal?

OS talks to process through signal

kill -l

9) SIGKILL signal kill->forceful 'don't kill system related process forceful'

15) SIGTERM(def.) signal termination->graceful

how to kill

kill <pid>

or

kill -15 <pid>

kill -9 <pid>

ex:

ps

kill 25661

jobs

[2]- **Terminated** sleep 10000

kill -9 25730

jobs

[1]+ **Killed** sleep 10000

killall <process name>

killall sleep

killall -9 sleep

Controlling Services and Daemons

INTRODUCTION TO systemd

The **systemd daemon** manages startup for Linux, including service startup and service management in general.

what is daemon:

daemons, system related background process either running or sleep at background. daemons mostly ended by 'd'

what is service:

in systemd sense service refers to a single daemon or multiple daemons that consider and control service life cycle.

to manage service, daemons use **systemctl** command.

```
# systemctl
```

to find out service daemon's use

```
# systemctl list-unit-files
# systemctl list-unit-files | grep "ssh"
sshd.service
# systemctl list-unit-files | grep "cron"
crond.service
# systemctl list-unit-files | grep "chrony"
chronyd.service

# systemctl       poweroff
                  reboot
                  hibernate
                  halt
                  disable
                  enable
                  is-active
                  is-enabled
                  is-failed
                  mask
                  unmask
                  status
                  start
                  stop
                  restart
                  reload
                  reload-or-restart
# systemctl status chronyd.service
# systemctl status sshd
# systemctl disable crond.service          ->disable means, service won't start at boot time. should start manually
# systemctl enable crond.service           ->enable means, service will start at boot time.
# systemctl is-active crond.service
# systemctl is-enabled crond.service
# systemctl is-failed crond.service
# systemctl start crond.service            ->start service or daemon
# systemctl stop crond.service             ->stop service or daemon
# systemctl restart crond.service          ->restart service or daemon after make change in configuration files should happens
# systemctl reload crond.service           ->reload service or daemon and it will effect just on changed not over whole service
# systemctl reload-or-restart crond.service ->ask linux to decide

# yum list autofs
# yum install autofs.x86_64 -y
# systemctl list-unit-files | grep "autofs"
autofs.service                    disabled
# systemctl status  autofs.service
● autofs.service - Automounts filesystems on demand
   Loaded: loaded (/usr/lib/systemd/system/autofs.service; disabled; vendor preset: disabled)
   Active: inactive (dead)
# systemctl enable autofs.service
# systemctl start autofs.service
```

restart vs reload

-restart means stop service/daemon and start service/daemon

-reload means just effect on changed not all service sections

mask and unmask

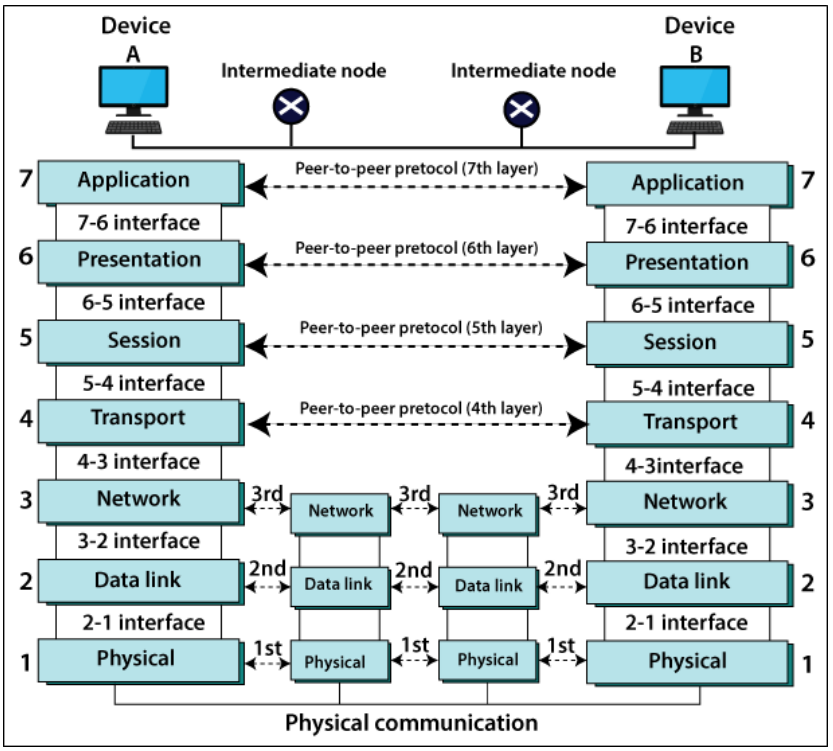
to prevent enable and start service/daemon by other services/daemons, mask it.

```
# systemctl disable crond.service
# systemctl stop  crond.service
# systemctl mask crond.service
# systemctl status crond.service
● crond.service
   Loaded: masked (Reason: Unit crond.service is masked.)
   Active: inactive (dead)
```

Managing Networking
what is network?
network means connect minimum 2hosts

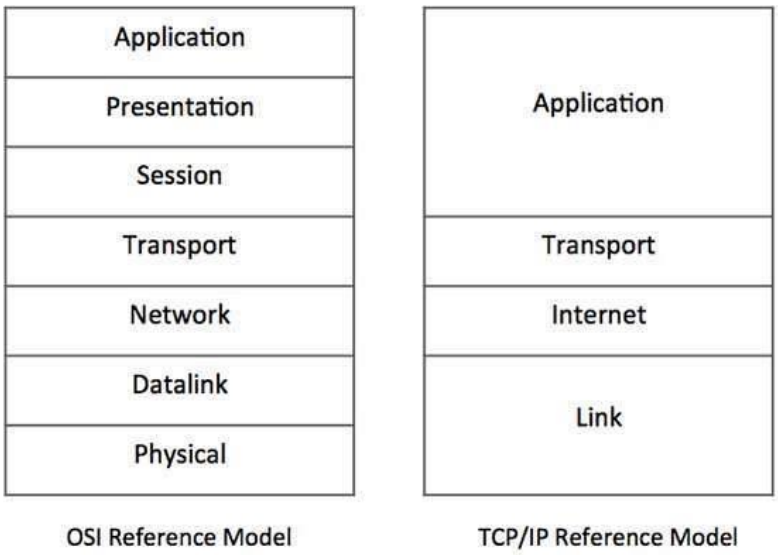
Network works based-on 2virtual model:
1-Open System Interconnection-OSI
2-Transmission Control Protocol/Internet Protocol-tcp/ip

1-OSI model
it works with 7layes:
7-Application
6-Presentation
5-Session
4-Transport
3-Network
2-Datalink
1-Physical



7-Application
closest layer to user.
6-Presentation
ensure data from one hosts accessible and readable by other host.
5-Session
initiate, manage and terminate session between hosts.
4-Transport
-output is segment
-TCP protocols
slow and reliable protocols like ssh, dns, http
-UDP protocols
fast and unreliable protocols like ntp
3-Network
-output is packet
-as device router, l3switch
-as protocols ipv4, ipv6
2-Datalink
-output is frame
-as device l2switch, bridge
-as protocol mas address
1-Physical
-as device hub
-output is signal in:
wire
	rj45
wireless
	wifi, wimax, lifi
light
	Fiber Optic

1-TCP/IP model
released by US-DOD



3-Network
-output is packet
-as device router, l3switch
-as protocols ipv4, ipv6

what is binary, decimal and hexadecimal units

binary	**decimal**	**hexadecimal**
0/1	0	0
on/off	1	1
	2	2
	3	3
	4	4
	5	5
	6	6
	7	7
	8	8
	9	9
		A 10
		B 11
		C 12
		D 13
		E 14
		F 15

smallest unit computer science, its **bit** it's **on/off 0/1**

1bit
4bits nibble
8bits byte/octet

7	**6**	**5**	**4**	**3**	**2**	**1**	**0**

2 power bit number

7	6	5	4	3	2	1	0
2	2	2	2	2	2	2	2
128	**64**	**32**	**16**	**8**	**4**	**2**	**1**

128+65+32+16+8+4+2+1=**255**

if whole bits be **on/1**->**255**

or
if whole bits be **off/0**->**0**

0	**0**	**0**	**0**	**0**	**0**	**0**	**0**

ex:
||00|000->200

what is ipv4

its 32bits in to 4byts/octet divided by **.** and each byte is between 0 to 255
|||||||.|||||||.|||||||.|||||||

ipv4 class
to better use ipv4 divided in to default 5classes

A0	to	126	
B128	to	191	
C192	to	223	
D224	to	239	multicasting
E240	to	255	R/D

NOTE:
-to detect ip's class check first octet from left with ip's classes
-127 uses for loopback interface, internal peruse
-0 uses to define network-id
-255 uses to define broadcast-ip

ipv4 comes in to 2types:
1-public, means has ping over internet network.
2-private, means doesn't have ping over internet network.

	A10.0.0.0	to	10.255.255.255
	B172.16.0.0	to	B172.31.255.255
	C192.168.0.0	to	192.168.255.255

ipv4 by default has 2parts
network part + host part
fixed changeable
A |||||||||.|||||||||.|||||||||.|||||||| most ipv4's
N H H H
B|||||||||.|||||||||.|||||||||.||||||||
N N H H
C|||||||||.|||||||||.|||||||||.|||||||| least ipv4's
N N N H

ipv4 example
192.168.50.12
10.1.1.1
172.25.36.20
180.56.25.63

when2ipv4 can connect direct to each other without router
1-should be in same CLASS
2-should has same network part

ex:
10.1.1.1.1 and 11.1.1.2 ->no direct ping
A A
172.25.10.1 and 172.25.12.25 ->yes, direct ping
B B

how machine will detect ip's class
humans:
compare with default classes
ex:
152.36.25.89 ->B class
machines:
through subnet mask.

what is subnet mask
defines ip's class for machines
who calculate
to calculate subnet mask Network part should be 1/on and Host part should be 0/off
A 255.0.0.0
B 255.255.0.0
C 255.255.255.0

192.168.1.1 255.255.255.0
172.25.250.10 255.255.0.0

NOTE: subnet mask decides about ip's behavior

ipv4 comes in to:
-classfull
standard ip and subnet mask
10.1.1.1 255.0.0.0
-classless
ip doesn't match with subnet mask
10.1.1.1 255.255.255.0

what is cidr/prefix
instead of **subnet mask** use **prefix**
to calculate prefix, **count each class network bits**
A/8
B/16
C/24
ex:
classless
10.1.1.1/24
10.1.1.1/8
10.1.1.1/16
classfull
10.1.1.1/8
172.25.250.10/16
192.168.100.1/24

NOTE: ip without subnet mask or prefix, should take it as **classfull**
150.41.23.125 ->B class

ipv4 Analyze
CLASS
Network-ID
First IP
Last IP
Broadcast IP
number of hosts

ex:
192.168.10.1/24

CLASS >C

Network-ID ->Host part should be 0
192.168.10.0

First IP ->network-id +1
192.168.10.1

Last IP ->Broadcast IP - 1
192.168.10.254

Broadcast IP -> Host part should be 1
192.168.10.255

number of available hosts
2^h-2
2^8-2=256-2=254
192.168.10.1 to 192.168.10.254

NOTE: don’t set network-id and broadcast-ip over nic card.

what is ipv6
it comes to 128bits in binary language. 8parts*16bits=128bits

ipv6 example
||||||||||||||||.||||||||||||||||.||||||||||||||||.||||||||||||||||.||||||||||||||||.||||||||||||||||.||||||||||||||||.||||||||||||||||

binary should concert to hexadecimal
0010000000000001:0000110110111000:0000101000001011:0001001011110000:0000000000000000:0000000000000000:0000000000000000:0000000000000001

ex:
[0000][1101][1011][1000]

[0000]	[1101]	[1011]	[1000]	
3 2 1 0 ->bits number	3 2 1 0	3 2 1 0		
3 2 1 0	8+4+0+1	8+0+2+1		
2 2 2 2	**13**	**11**	**8**	->**0DB8**

0

after convert to Hexadecimal
2001:**0db8**:**0a0b**:12f0:0000:0000:0000:**0001**

3rules to make ipv6 better
1-dicard leading 0
2001:**0db8**:**0a0b**:12f0:0000:0000:0000:**1**
to
2001:**db8**:**a0b**:12f0:0000:0000:0000:1

2-if 2 or more 0block coming after each, remove 0blocks and put ::
2001:db8:a0b:12f0:**0000**:**0000**:**0000**:1
to
2001:db8:a0b:12f0**::**1

3-replace 0000 to 0
2001:db8:a0b:12f0**::0000**
to
2001:db8:a0b:12f0**::0**

ipv6 type
1-global-unicast	->ipv6 public, it has ping on internet	ipv4public == ipv6 public
2-link-local	->ipv6 private, doesn't ping on internet	ipv4private == ipv6 private
3-unique-local	->in public network acts as ipv6 public and in private network acts as private ipv6	

	ipv4	ipv6
local host	127.0.0.0	::1/128
default route	0.0.0./0	::
global-unicast		2000::/3
link-local		fe80::/64
unique-local		fd00::/8