

Operating Systems — Linux and Lightweight kernels

James E. Prewett

October 8, 2008

Popular Linux Distributions

RedHat and RedHat clones

Environment Modules

Compilers

Essential Linux commands

A note on “randomness”

Lightweight Kernels

Popular Linux Distributions

This list is by no means complete

- ▶ RedHat
- ▶ Fedora
- ▶ Scientific Linux
- ▶ CentOS
- ▶ SuSE/SLES
- ▶ OpenSuSE
- ▶ Debian
- ▶ Ubuntu
- ▶ Gentoo

RedHat and RedHat clones:

You can never have too much of a good thing!

Package Manager:	RPM
Package Format:	RPM

What ***ARE*** CentOS, Scientific Linux, and Fedora?

When to pick RedHat over one of its clones:

Pick RedHat when you have:

- ▶ Plenty of budget for the licenses

When to pick RedHat over one of its clones:

Pick RedHat when you have:

- ▶ Plenty of budget for the licenses
- ▶ Support concerns

When to pick RedHat over one of its clones:

Pick RedHat when you have:

- ▶ Plenty of budget for the licenses
- ▶ Support concerns
- ▶ 3rd Party Support concerns (Oracle, etc.)

When to pick RedHat over one of its clones:

Pick RedHat when you have:

- ▶ Plenty of budget for the licenses
- ▶ Support concerns
- ▶ 3rd Party Support concerns (Oracle, etc.)
- ▶ ... to make your manager sleep better ;)

When to pick a RedHat clone:

- ▶ Pick any of the clones to save some money!

When to pick a RedHat clone:

- ▶ Pick any of the clones to save some money!
- ▶ Pick Fedora if you want the latest in the RedHat world

When to pick a RedHat clone:

- ▶ Pick any of the clones to save some money!
- ▶ Pick Fedora if you want the latest in the RedHat world
- ▶ Pick CentOS if you want a (free!) rebuild of RHEL

When to pick a RedHat clone:

- ▶ Pick any of the clones to save some money!
- ▶ Pick Fedora if you want the latest in the RedHat world
- ▶ Pick CentOS if you want a (free!) rebuild of RHEL
- ▶ Pick Scientific Linux if you want a (free!) rebuild of RHEL with a bit of a “scientific computing” bent to it.

When to pick a RedHat clone:

- ▶ Pick any of the clones to save some money!
- ▶ Pick Fedora if you want the latest in the RedHat world
- ▶ Pick CentOS if you want a (free!) rebuild of RHEL
- ▶ Pick Scientific Linux if you want a (free!) rebuild of RHEL with a bit of a “scientific computing” bent to it.
- ▶ Pick any to be simpler to maintain than official RedHat IMHO :P

When to pick SuSE Enterprise Server:

Pick SLES when you have:

- ▶ Plenty of budget for the licenses
Less budget required than RedHat!

... Or, choose OpenSuSE to save some cash (and, IMO, some headache!)

When to pick SuSE Enterprise Server:

Pick SLES when you have:

- ▶ Plenty of budget for the licenses
Less budget required than RedHat!
- ▶ Support concerns

... Or, choose OpenSuSE to save some cash (and, IMO, some headache!)

When to pick SuSE Enterprise Server:

Pick SLES when you have:

- ▶ Plenty of budget for the licenses
Less budget required than RedHat!
- ▶ Support concerns
- ▶ 3rd Party Support concerns (Oracle, etc.)

... Or, choose OpenSuSE to save some cash (and, IMO, some headache!)

When to pick SuSE Enterprise Server:

Pick SLES when you have:

- ▶ Plenty of budget for the licenses
Less budget required than RedHat!
- ▶ Support concerns
- ▶ 3rd Party Support concerns (Oracle, etc.)
- ▶ ... to make your manager sleep better ;)

... Or, choose OpenSuSE to save some cash (and, IMO, some headache!)

Other Popular Linux Distributions

- ▶ Debian Gnu/Linux – A very conservative stability oriented distribution. Installing and upgrading packages is simple, but graphical tools are lacking.
- ▶ Ubuntu Linux – Based on Debian. Timely releases. Focus on a nice user desktop. “Meant to compliment Debian”.
- ▶ Gentoo Linux – Portage system inspired by FreeBSD Ports Tree. Pretty much the entire system is compiled (on your system) to be optimized for your hardware.

Environment Modules

Environment Modules provide a convenient, consistent way to modify a user's environment to enable the useage of a library, application, or piece of documentation.

Modules can:

- ▶ Set/Unset environment variables

Environment Modules

Environment Modules provide a convenient, consistent way to modify a user's environment to enable the useage of a library, application, or piece of documentation.

Modules can:

- ▶ Set/Unset environment variables
- ▶ Add-to/Remove from PATHs & MANPATHs, etc.

Environment Modules

Environment Modules provide a convenient, consistent way to modify a user's environment to enable the useage of a library, application, or piece of documentation.

Modules can:

- ▶ Set/Unset environment variables
- ▶ Add-to/Remove from PATHs & MANPATHs, etc.
- ▶ be loaded and *unloaded* dynamically

Environment Modules

Environment Modules provide a convenient, consistent way to modify a user's environment to enable the useage of a library, application, or piece of documentation.

Modules can:

- ▶ Set/Unset environment variables
- ▶ Add-to/Remove from PATHs & MANPATHs, etc.
- ▶ be loaded and *unloaded* dynamically
- ▶ be used to manage different versions of software

Environment Modules

Environment Modules provide a convenient, consistent way to modify a user's environment to enable the useage of a library, application, or piece of documentation.

Modules can:

- ▶ Set/Unset environment variables
- ▶ Add-to/Remove from PATHs & MANPATHs, etc.
- ▶ be loaded and *unloaded* dynamically
- ▶ be used to manage different versions of software
- ▶ be bundled into “meta-modules” to load complex sets of software

Environment Modules

Environment Modules provide a convenient, consistent way to modify a user's environment to enable the useage of a library, application, or piece of documentation.

Modules can:

- ▶ Set/Unset environment variables
- ▶ Add-to/Remove from PATHs & MANPATHs, etc.
- ▶ be loaded and *unloaded* dynamically
- ▶ be used to manage different versions of software
- ▶ be bundled into “meta-modules” to load complex sets of software
- ▶ be used by all popular shells:
bash, ksh, zsh, sh, csh, tcsh, as well as some scripting languages such as perl

Using Environment Modules

First, we'll load the module for GCC 3.4.6:

```
$ module load gcc/3.4.6  
$ which gcc  
/opt/gcc-3.4.6/bin/gcc
```

Now, we'll switch to the module for GCC 4.1.2:

```
$ module load gcc/4.1.2  
$ which gcc  
/usr/bin/gcc
```

Now, we'll unload the module:

```
$ module unload gcc  
$ which gcc  
gcc not found
```

Popular Compilers & Languages

Compiler Vendor	Language(s)
GCC	C, C++, Objective-C, Fortran, Java, Ada
INTEL	C, C++, Fortran
Portland Group (PGI)	C, C++, Fortran
PathScale	C, C++, Fortran
IBM XLC	C, C++
IBM XLF	Fortran
NAG	Fortran

Popular Compilers & Supported Processors

Compiler Vendor	Processor(s)
GCC	... A lot ...
INTEL	INTEL
Portland Group (PGI)	x86, x86-64
PathScale	x86, x86-64, AMD64, EM64T
IBM XLC	Power Series, (Incl. PPC)
IBM XLF	Power Series, (Incl. PPC)
NAG	Several

Popular Compilers Advantages

Compiler Vendor	Advantages
GCC	Many platforms, No cost
INTEL	Heavily Optimized for INTEL Hardware
Portland Group (PGI)	Good x86, x86-64 performance
PathScale	Good 64 bit performance
IBM XLC	Heavily Optimized on Power processors
IBM XLF	Heavily Optimized on Power processors
NAG	<i>Great</i> for debugging!

Essential Linux commands

- ▶ top(1)
- ▶ ps(1)
- ▶ lsof(8)
- ▶ kill(1)
- ▶ df(1)

top output

```
top - 11:28:13 up 15 days, 1:47, 1 user, load average: 4.01, 4.01, 4.00
Tasks: 85 total, 5 running, 80 sleeping, 0 stopped, 0 zombie
Cpu(s):100.0%us, 0.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 8308224k total, 6491856k used, 1816368k free, 48616k buffers
Swap: 2104472k total, 0k used, 2104472k free, 6225268k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
20375	dgxu	25	0	469m	21m	2036	R	100	0.3	837:48.90	c32a2.exe
20376	dgxu	25	0	469m	21m	2036	R	100	0.3	837:48.95	c32a2.exe
20377	dgxu	25	0	469m	21m	2036	R	100	0.3	837:37.18	c32a2.exe
20378	dgxu	25	0	469m	21m	2036	R	100	0.3	837:48.37	c32a2.exe
1	root	16	0	720	280	244	S	0	0.0	0:02.45	init
2	root	RT	0	0	0	0	S	0	0.0	0:00.04	migration/0
3	root	34	19	0	0	0	S	0	0.0	0:00.00	ksoftirqd/0
4	root	RT	0	0	0	0	S	0	0.0	0:00.00	migration/1
5	root	34	19	0	0	0	S	0	0.0	0:00.00	ksoftirqd/1
6	root	RT	0	0	0	0	S	0	0.0	0:00.00	migration/2
7	root	34	19	0	0	0	S	0	0.0	0:00.00	ksoftirqd/2
8	root	RT	0	0	0	0	S	0	0.0	0:00.00	migration/3
9	root	34	19	0	0	0	S	0	0.0	0:00.00	ksoftirqd/3
10	root	10	-5	0	0	0	S	0	0.0	0:00.12	events/0
11	root	10	-5	0	0	0	S	0	0.0	0:00.00	events/1
12	root	10	-5	0	0	0	S	0	0.0	0:00.00	events/2
13	root	10	-5	0	0	0	S	0	0.0	0:00.08	events/3

ps output

```
download@nano31:~$ ps auxwww | grep -v root | grep -v download
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
100        2599  0.0  0.0   3416   988 ?        Ss   Sep17   0:01 /usr/bin/dbus-daemon --system
nobody    3157  0.0  0.0   1556   424 ?        Ss   Sep17   0:00 /sbin/portmap
daemon    3269  0.0  0.0   3252   916 ?        Ss   Sep17   0:00 /usr/sbin/slpd
ntp       3975  0.0  0.0   4164  4164 ?        SLs  Sep17   0:00 /usr/sbin/ntpd -p /var/lib/ntp/var/run/ntp
postfix   4118  0.0  0.0   5412  1672 ?        S    Sep17   0:00 qmgr -l -t fifo -u
dgxu     20279  0.0  0.0   4832  2004 ?        Ss   Oct01   0:00 -csh
dgxu     20331  0.0  0.0   1844   612 ?        S    Oct01   0:00 pbs_demux
dgxu     20370  0.0  0.0   4372  1664 ?        S    Oct01   0:00 /usr/bin/csh /var/spool/torque/mom_priv/
dgxu     20375  99.9  0.2 480564 21920 ?        R    Oct01  946:34 c32a2.exe
dgxu     20376  99.9  0.2 480576 21940 ?        R    Oct01  946:34 c32a2.exe
dgxu     20377  99.9  0.2 480576 21940 ?        R    Oct01  946:22 c32a2.exe
dgxu     20378  99.9  0.2 480568 21940 ?        R    Oct01  946:31 c32a2.exe
postfix   21805  0.0  0.0   5376  1644 ?        S    13:03   0:00 pickup -l -t fifo -u
```

ls -lsof lists open files

- ▶ Currently open files

ls -lsof lists open files

- ▶ Currently open files

lsof lists open files

- ▶ Currently open files
- ▶ Open Network connections — *-i*

lsof lists open files

- ▶ Currently open files
- ▶ Open Network connections — *-i*
- ▶ Open files in a given directory — *+d <directory>*

ls -lsof lists open files

- ▶ Currently open files
- ▶ Open Network connections — *-i*
- ▶ Open files in a given directory — *+d <directory>*
- ▶ Open NFS files — *-N*

ls -lsof lists open files

- ▶ Currently open files
- ▶ Open Network connections — *-i*
- ▶ Open files in a given directory — *+d <directory>*
- ▶ Open NFS files — *-N*
- ▶ Unix Domain Sockets (used for IPC, etc.) — *-U*

ls -lsof lists open files

- ▶ Currently open files
- ▶ Open Network connections — *-i*
- ▶ Open files in a given directory — *+d <directory>*
- ▶ Open NFS files — *-N*
- ▶ Unix Domain Sockets (used for IPC, etc.) — *-U*
- ▶ a bunch of other options... RTFM!

Without any arguments, lsof lists all open files on the system

```
nano:~ # lsof | head -25
```

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE	NODE NAME
init	1	root	cwd	DIR	8,2	696	2 /
init	1	root	rtd	DIR	8,2	696	2 /
init	1	root	txt	REG	8,2	517716	31071 /sbin/init
init	1	root	mem	REG	0,0		0 [heap] (stat: No such file or di
init	1	root	10u	FIFO	0,14		2550 /dev/initctl
migration	2	root	cwd	DIR	8,2	696	2 /
migration	2	root	rtd	DIR	8,2	696	2 /
migration	2	root	txt	unknown			/proc/2/exe
ksoftirqd	3	root	cwd	DIR	8,2	696	2 /
ksoftirqd	3	root	rtd	DIR	8,2	696	2 /
ksoftirqd	3	root	txt	unknown			/proc/3/exe
migration	4	root	cwd	DIR	8,2	696	2 /
migration	4	root	rtd	DIR	8,2	696	2 /
migration	4	root	txt	unknown			/proc/4/exe
ksoftirqd	5	root	cwd	DIR	8,2	696	2 /
ksoftirqd	5	root	rtd	DIR	8,2	696	2 /
ksoftirqd	5	root	txt	unknown			/proc/5/exe
migration	6	root	cwd	DIR	8,2	696	2 /
migration	6	root	rtd	DIR	8,2	696	2 /
migration	6	root	txt	unknown			/proc/6/exe
ksoftirqd	7	root	cwd	DIR	8,2	696	2 /
ksoftirqd	7	root	rtd	DIR	8,2	696	2 /
ksoftirqd	7	root	txt	unknown			/proc/7/exe
migration	8	root	cwd	DIR	8,2	696	2 /

Isof -i output

Have Isof list open Network “files”

```
nano:~ # lsof -i | head -25
```

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE	NODE	NAME
ipmitool	1092	root	4u	IPv4	936203		UDP	nano.nano.alliance.unm.edu:10422->nano16-admin.na
ipmitool	1689	root	4u	IPv4	838800		UDP	nano.nano.alliance.unm.edu:4625->nano04-admin.na
conserver	2786	root	3u	IPv4	7037		TCP	*:console (LISTEN)
conserver	2790	root	3u	IPv4	6386		TCP	*:47546 (LISTEN)
maui	2825	root	5u	IPv4	6531		TCP	*:42559 (LISTEN)
maui	2825	root	6u	IPv4	6532		TCP	*:42560 (LISTEN)
maui	2825	root	7u	IPv4	22318441		TCP	nano.nano.alliance.unm.edu:28955->nano.nano.allia
maui	2825	root	8u	IPv4	22318455		TCP	*:pbs_sched (LISTEN)
conserver	2833	root	3u	IPv4	6530		TCP	*:47591 (LISTEN)
ipmitool	3425	root	4u	IPv4	943140		UDP	nano.nano.alliance.unm.edu:11023->nano17-admin.na
sshd	4231	root	3u	IPv6	594518		TCP	nano.alliance.unm.edu:ssh->ycg34884vig.dl.ac.uk:5
sshd	4233	gbassi	3u	IPv6	594518		TCP	nano.alliance.unm.edu:ssh->ycg34884vig.dl.ac.uk:5
lmgrd	4358	root	0u	IPv4	1250597		TCP	*:27000 (LISTEN)
lmgrd	4358	root	3u	IPv4	1250621		TCP	localhost:27000->localhost:12969 (ESTABLISHED)
atomist	4359	root	0u	IPv4	1250597		TCP	*:27000 (LISTEN)
atomist	4359	root	3u	IPv4	1250600		TCP	*:18965 (LISTEN)
atomist	4359	root	5u	IPv4	1250620		TCP	localhost:12969->localhost:27000 (ESTABLISHED)
atomist	4359	root	16u	IPv4	2561344		TCP	nano.nano.alliance.unm.edu:18965->nano.nano.allia
ipmitool	4985	root	4u	IPv4	848505		UDP	nano.nano.alliance.unm.edu:5366->nano05-admin.na
sshd	5331	root	3u	IPv6	1276944		TCP	nano.alliance.unm.edu:ssh->augerdata1.phys.unm.edu
sshd	5333	bbecker	3u	IPv6	1276944		TCP	nano.alliance.unm.edu:ssh->augerdata1.phys.unm.edu
sshd	5333	bbecker	7u	IPv4	1277133		TCP	localhost:6013 (LISTEN)
sshd	5333	bbecker	8u	IPv6	1277134		TCP	localhost:6013 (LISTEN)
ipmitool	5345	root	4u	IPv4	948946		UDP	nano.nano.alliance.unm.edu:11175->nano18-admin.na

lsfd +d /tmp output

Have lsfd list open files in a directory

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE	NODE	NAME
gdm	12029	root	6u	unix	0xf4c8be40		48458	/tmp/.gdm_socket
bash	13447	download	cwd	DIR	8,2	72	942748	/tmp/foo
emacs	18184	download	cwd	DIR	8,2	72	942748	/tmp/foo
sbcl	18193	download	cwd	DIR	8,2	72	942748	/tmp/foo

Isof -N output

Have Isof list open NFS files

```
nano:~ # lsof -N | head -25
```

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE	NODE	NAME
tcsh	4235	gbassi	cwd	DIR	0,19	4096	20480512	/users/gbassi/CSR_NANO/300lambda (serrano.a
tcsh	5335	bbecker	cwd	DIR	0,22	21408	312134	/nano/scratch/bbecker/anisop/DATA_Box (nan
tcsh	6028	dianah	cwd	DIR	0,19	4096	35405932	/users/dianah (serrano.alliance.unm.edu:/e
tcsh	6129	dianah	cwd	DIR	0,19	4096	35405932	/users/dianah (serrano.alliance.unm.edu:/e
sftp-serv	6151	dianah	cwd	DIR	0,19	4096	35405932	/users/dianah (serrano.alliance.unm.edu:/e
tcsh	9405	gsmith	cwd	DIR	0,21	4096	89833556	/nfs/scratch/gsmith/blact/L3/dyn (serrano.a
tcsh	10241	erbb123	cwd	DIR	0,19	4096	886392	/users/erbb123/SNL/R2LT/Run10 (serrano.all
tcsh	15753	bbecker	cwd	DIR	0,22	21408	312134	/nano/scratch/bbecker/anisop/DATA_Box (nan
vi	18238	dianah	cwd	DIR	0,19	4096	35405932	/users/dianah (serrano.alliance.unm.edu:/e
vi	18238	dianah	4u	REG	0,19	16384	122110610	/users/dianah/.opt.out.swp (serrano.allian
tcsh	18501	dianah	cwd	DIR	0,19	4096	35405932	/users/dianah (serrano.alliance.unm.edu:/e
bash	18665	download	cwd	DIR	0,19	8192	66322440	/users/download (serrano.alliance.unm.edu:/
tcsh	18811	jsegroup	cwd	DIR	0,19	4096	45105785	/users/jsegroup/tomas/compile (serrano.all
vnl	20496	dianah	cwd	DIR	0,19	4096	35405932	/users/dianah (serrano.alliance.unm.edu:/e
vnl_exec	20498	dianah	cwd	DIR	0,19	4096	35405932	/users/dianah (serrano.alliance.unm.edu:/e
vnl_exec	20498	dianah	5w	REG	0,19	0	121913498	/users/dianah/.vnl/vnl.log (serrano.allian
vnl_exec	20498	dianah	9r	REG	0,19	79515	35423375	/users/dianah/.vnl/saves/2_0_1/1220647684.y
vnl_exec	20498	dianah	11r	REG	0,19	11026	122110623	/users/dianah/ada1.vnl (serrano.alliance.un
tcsh	27220	gsmith	cwd	DIR	0,21	4096	2113589	/nfs/scratch/gsmith/ospf/spvc_mm1 (serrano
tcsh	30608	dianah	cwd	DIR	0,19	4096	35405932	/users/dianah (serrano.alliance.unm.edu:/e
tcsh	30961	gsmith	cwd	DIR	0,19	4096	50698	/users/gsmith (serrano.alliance.unm.edu:/e
sftp-serv	30983	gsmith	cwd	DIR	0,19	4096	50698	/users/gsmith (serrano.alliance.unm.edu:/e
tcsh	31521	bbecker	cwd	DIR	0,19	4096	33210	/users/bbecker (serrano.alliance.unm.edu:/
tcsh	31685	bbecker	cwd	DIR	0,19	4096	33210	/users/bbecker (serrano.alliance.unm.edu:/

Isof -U output

Have Isof list open UNIX domain sockets (used for IPC, etc.)

```
nano:~ # lsof -U | head -25
```

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE	NODE	NAME
udev	1115	root	3u	unix	0xdff57c80		2704	socket
resmgrd	2766	root	3u	unix	0xdff57580		6309	/var/run/.resmgr_socket
dbus-daem	2787	messagebus	3u	unix	0xdff573c0		6367	/var/run/dbus/system_bus_socket
dbus-daem	2787	messagebus	6u	unix	0xdff57740		6381	socket
dbus-daem	2787	messagebus	7u	unix	0xdff57900		6382	socket
dbus-daem	2787	messagebus	8u	unix	0xf596b580		19971	/var/run/dbus/system_bus_socket
acpid	2792	root	4u	unix	0xdff57200		6403	/var/run/acpid.socket
acpid	2792	root	5u	unix	0xf6d33200		15147	/var/run/acpid.socket
acpid	2792	root	7u	unix	0xf52fb580		48677	/var/run/acpid.socket
acpid	2792	root	8u	unix	0xf4c8bc80		48678	socket
hald	3108	root	7u	unix	0xdff57ac0		7493	socket
hald	3108	root	8u	unix	0xdff57040		7494	socket
hald	3108	root	9u	unix	0xf7de1040		7495	socket
hald	3108	root	11u	unix	0xdff57e40		15865	socket
hald	3108	root	12u	unix	0xf596b740		19970	socket
hald	3108	root	13u	unix	0xf596bac0		19621	socket
sshd	4231	root	5u	unix	0xf7a18200		594625	socket
sshd	4233	gbassi	4u	unix	0xf6d333c0		594624	socket
hald-addo	4830	root	3u	unix	0xf7de1c80		15144	socket
hald-addo	4830	root	4u	unix	0xf6d33040		15146	socket
sshd	5331	root	5u	unix	0xf4c8b200		1277100	socket
sshd	5333	bbecker	4u	unix	0xf37b53c0		1277099	socket
sshd	6020	root	5u	unix	0xc7422040		20698874	socket
sshd	6026	dianah	4u	unix	0xc7422580		20698873	socket

kill(1)

kill -9 kills processes *dead*

Use kill for, well, what it says.. to kill processes!

kill can also be used to send an arbitrary signal, such as SIGHUP or SIGUSR to a process.

df output

```
nano:~ # df
Filesystem            1K-blocks      Used Available Use% Mounted on
/dev/sda2              76017196  53164756  22852440   70% /
udev                  4154112      116    4153996    1% /dev
serrano.alliance.unm.edu:/export/home/alliance
1007930816 956196432    534432 100% /users
serrano.alliance.unm.edu:/nfs/scratch
960412336 910471520   1154624 100% /nfs/scratch
nanoserv.nano.alliance.unm.edu:/raid
3165816480 2180893184 984923296 69% /nano/scratch
```

A note on “Randomness”

How is /dev/random populated (in Linux)? Where does it get its **entropy** from?

- ▶ Disk interrupts
- ▶ Keyboard interrupts
- ▶ Mouse interrupts
- ▶ Internal Hardware Random Number Generators
Lucky you!
- ▶ ****THATS IT****

How do I see how much randomness is available?

`/dev/random` is blocking on me!

- ▶ `/proc/sys/kernel/random/entropy_avail` — available entropy (more is good!)

How do I see how much randomness is available?

`/dev/random` is blocking on me!

- ▶ `/proc/sys/kernel/random/entropy_avail` — available entropy (more is good!)
- ▶ `/proc/sys/kernel/random/read_wakeup_threshold` — when bytes will be available

How do I see how much randomness is available?

`/dev/random` is blocking on me!

- ▶ `/proc/sys/kernel/random/entropy_avail` — available entropy (more is good!)
- ▶ `/proc/sys/kernel/random/read_wakeup_threshold` — when bytes will be available
- ▶ `/proc/sys/kernel/random/write_wakeup_threshold` — when the kernel will try to start collecting more entropy

How do I see how much randomness is available?

`/dev/random` is blocking on me!

- ▶ `/proc/sys/kernel/random/entropy_avail` — available entropy (more is good!)
- ▶ `/proc/sys/kernel/random/read_wakeup_threshold` — when bytes will be available
- ▶ `/proc/sys/kernel/random/write_wakeup_threshold` — when the kernel will try to start collecting more entropy
- ▶ What if I **NEVER** get a larger number in `entropy_avail`?
... and therefore `/dev/random` blocks forever?!!!

How do I see how much randomness is available?

/dev/random is blocking on me!

- ▶ /proc/sys/kernel/random/entropy_avail — available entropy (more is good!)
- ▶ /proc/sys/kernel/random/read_wakeup_threshold — when bytes will be available
- ▶ /proc/sys/kernel/random/write_wakeup_threshold — when the kernel will try to start collecting more entropy
- ▶ What if I **NEVER** get a larger number in entropy_avail? ... and therefore /dev/random blocks forever?!!!
- ▶ About all you can do (under Linux) is **rng-tools**

How do I see how much randomness is available?

`/dev/random` is blocking on me!

- ▶ `/proc/sys/kernel/random/entropy_avail` — available entropy (more is good!)
- ▶ `/proc/sys/kernel/random/read_wakeup_threshold` — when bytes will be available
- ▶ `/proc/sys/kernel/random/write_wakeup_threshold` — when the kernel will try to start collecting more entropy
- ▶ What if I **NEVER** get a larger number in `entropy_avail`?
... and therefore `/dev/random` blocks forever?!!!
- ▶ About all you can do (under Linux) is **rng-tools**
- ▶ `rng-tools` allows you to “seed” `/dev/random` using `/dev/urandom`

Lightweight Kernels

Lightweight Kernels were developed after observing that:

- ▶ Most applications have no need for most UNIX processes

Lightweight Kernels

Lightweight Kernels were developed after observing that:

- ▶ Most applications have no need for most UNIX processes
- ▶ General-purpose multiprocessing activity gets in the way of compute jobs

Lightweight Kernels

Lightweight Kernels were developed after observing that:

- ▶ Most applications have no need for most UNIX processes
- ▶ General-purpose multiprocessing activity gets in the way of compute jobs
- ▶ Process scheduling gets in the way of compute jobs

Lightweight Kernels

Lightweight Kernels were developed after observing that:

- ▶ Most applications have no need for most UNIX processes
- ▶ General-purpose multiprocessing activity gets in the way of compute jobs
- ▶ Process scheduling gets in the way of compute jobs
- ▶ The above combined on **MANY** machines can *destroy* your performance!