

アイランドセンター中嶋事務所

## これだけは知っていれば立派な管理者になれるマニュアル

※SUSE Enterprise Server 10x/11 を基本としています。

## SUSE Linux のシステム起動、シャットダウン、リスタート、ログアウト

### ログイン

```
vm1 login: root
Password: yourpassword
Last login: Wed Oct 22 13:11:03 JST 2008 on xvc0
vm1:~ # <--- プロンプトが表示されます
```

### ログアウト(exit, ctrl+D)

```
vm1:/tmp/vm1 # exit (または Ctrl+D)
logout
```

```
Welcome to SUSE Linux Enterprise Server 10 SP2 (x86_64) - Kernel
2.6.16.60-0.21-xen (xvc0).
```

```
vm1 login: <--- 再びログインプロンプトが出ます
```

### 再起動(リブート)

```
vm1:~ # shutdown -r 0 ( 0 秒後に Reboot を開始 )
```

```
Broadcast message from root (xvc0) (Thu Oct 23 15:49:27 2008):
```

```
The system is going down for reboot NOW!
```

### シャットダウン(電源を落とす)

```
vm1:~ # shutdown -h 0 ( 0 秒後に Halt-停止を開始 )
```

```
Broadcast message from root (xvc0) (Thu Oct 23 15:49:27 2008):
```

```
The system is going down for reboot NOW!
```

## コマンド入力の省略[tab]

ターミナル(端末)から、コマンドプロンプトを開いてコマンドを入力するとき、入力が楽に行えるようにコマンドの補完機能があります。つづりがよく判らないファイルを探す時に **Tab** キーを押します。

とにかく「**コマンドラインが長くて複雑だ**」と感じた時は **tab** キーを押すと解決できることがあります。

```
vm1:~ # cd /e[Tab]tc[Enter] <- cd /e で[tab]を押すと tc が補完されます。
vm1:/etc #
vm1:/etc #
```

また、**Tab** キーを2度押しすると、候補がリストアップされます。

```
vm1:/etc # cd s[Tab][Tab] <- s で始まる候補がリストアップ
samba/      skel/      snmp/      ssl/      sysconfig/
security/   slp.reg.d/  ssh/      susehelp.d/ syslog-ng/
vm1:/etc # cd se[Tab]curity[Enter] <- se[tab]を押すと security がリストアップ
vm1:/etc/security #
```

## コマンドヒストリー

コマンドプロンプトから上下矢印キーで、過去に実行したコマンドヒストリーを呼び出すことができます。

```
vm1:~/Desktop # /etc/init.d/apache2 stop
Shutting down httpd2 (waiting for all children to terminate)      done
vm1:~/Desktop # /etc/init.d/apache2 start <- 上矢印キーを押して stop を start に置き換える
Starting httpd2 (prefork)                                          done
vm1:~/Desktop #
```

## 基本的なファイル操作

### ディレクトリの確認( cd ,pwd, ls )

```

vm1:~ # pwd <- 現在いる場所を確認
/root
vm1:~ #
vm1:~ # cd /etc/init.d/ <- 移動する
vm1:/etc/init.d # pwd
/etc/init.d <- 移動先を確認
vm1:/etc/init.d #
vm1:/etc/init.d # ls <- ディレクトリの内容を確認
.depend.boot          boot.md              ipmi                 rc2.d
.depend.start         boot.multipath      ipxmount            rc3.d
.depend.stop          boot.open-iscsi     irq_balancer        rc4.d
Makefile              boot.proc           joystick            rc5.d

: 中略

vm1:/etc/init.d # cd <- 単に cd と打つ場合
vm1:~ # pwd
/root <-- ホームディレクトリに戻る
vm1:~ #

```

### ディレクトリ内容の確認( ls )

```

vm1:/home/nakaj/Documents/myfile # ls <- 一覧を表示
my.txt

vm1:/home/nakaj/Documents/myfile # ls -l <- 一覧をリスト表示
total 4
-rw-r--r-- 1 root root 143 Oct 29 22:50 my.txt

vm1:/home/nakaj/Documents/myfile # ls -al <- 隠しファイルを含めリスト表示
total 4
drwxr-xr-x 2 root root 72 Oct 29 22:50 .
drwx----- 3 nakaj users 104 Oct 29 22:50 ..
-rw-r--r-- 1 root root 143 Oct 29 22:50 my.txt

vm1:/home/nakaj/Documents/myfile # ls -alu <- 最終アクセス時刻を表示
total 4
drwxr-xr-x 2 root root 72 Oct 29 22:51 .
drwx----- 3 nakaj users 104 Oct 29 22:50 ..
-rw-r--r-- 1 root root 143 Oct 29 22:52 my.txt

```

## ファイルを探す(find)

現在のディレクトリ(ドット) 以下にある “csrf” で始まる全てのファイルを探す

```
vm1:/srv/www # find . -name "csrf*.*"
./cgi-bin/cbvm1/code/include/fw/csrf.csp
./cgi-bin/cbvm1/code/include/fw/csrf_list.csp
./cgi-bin/cbvm1/page.old/vm1/csrf_script.tpl
vm1:/srv/www #
```

/ (ルート) から全ての named.conf を探す

```
xenhost:~ # find / -name "named.conf"
/etc/named.conf
/var/lib/named/etc/named.conf
/usr/share/doc/packages/bind/config/ISC-examples/named.conf
xenhost:~ #
```

※ヒント 画面に収まり切らない場合は **command | more** で画面スクロールを止めます。

ex) # find / -name "a\*" | more <--- a がある全てのファイル

※ヒント 内容をテキストファイルに書き出す(/ ルートから “a\*” で始まる全てのファイルを検索してテキストデータを作る。

```
vm1:~ # find / -name "a*" > from-a.txt
vm1:~ # cat from-a.txt | more
/bin/ash
/bin/awk
/bin/arch
/bin/ash.static
: 以下略
```

## コピー、移動、削除

### コピー(cp)

```
vm1:/home/nakaj/myfile # cp -r -v /root/backup/mysql-5.0/* .
/root/backup/mysql-5.0/bin' -> `./bin'
/root/backup/mysql-5.0/bin/mysql' -> `./bin/mysql'
/root/backup/mysql-5.0/bin/mysqladmin' -> `./bin/mysqladmin'
/root/backup/mysql-5.0/bin/mysqlcheck' -> `./bin/mysqlcheck'
/root/backup/mysql-5.0/bin/mysqlbinlog' -> `./bin/mysqlbinlog'
: 中略'
vm1:/home/nakaj/myfile #
```

※ □□ cp “オプション” “コピー元” “コピー先”  
上の例では **-r** (サブディレクトリを含む) **-v** (コピーの詳細を表示) コピー元 (絶対パス) . (今のディレクトリ) へのコピーです。

※ **-i** 上書き確認

## 削除(rm)

```
vm1:/home/nakaj/Documents/myfile # ls -al
total 0
drwxr-xr-x 5 root root 120 Oct 29 17:45 .
drwx----- 4 nakaj users 152 Oct 29 17:31 ..
drwxr-xr-x 2 root root 328 Oct 29 17:45 bin <- ディレクトリを含めて沢山のファイルがある
drwxr-xr-x 3 root root 72 Oct 29 17:45 data
drwxr-xr-x 4 root root 144 Oct 29 17:45 share
vm1:/home/nakaj/Documents/myfile #
vm1:/home/nakaj/Documents/myfile # rm * -r
vm1:/home/nakaj/Documents/myfile #
vm1:/home/nakaj/Documents/myfile # ls -al
total 0
drwxr-xr-x 2 root root 48 Oct 29 18:09 .
drwx----- 4 nakaj users 152 Oct 29 17:31 ..
vm1:/home/nakaj/Documents/myfile #
```

※□ **rm \* -r** は現在のディレクトリ以下全て削除します。**確認メッセージがないので充分注意**してください。

※ **rm \* -r -i** (削除確認付き)

※ **rm \* -rf** (**f** は削除できないファイルへの強制オプション、使う場合は充分注意すること)

## ディレクトリの作成と削除 (mkdir, rmdir)

```
vm1:/home/nakaj/Documents # ls -al
total 5
drwx----- 2 nakaj users   80 Oct 29 22:44 .
drwxr-xr-x  8 nakaj users  584 May 28 16:13 ..
-rw-r--r--  1 nakaj users 1119 May 28 16:13 .directory

vm1:/home/nakaj/Documents # mkdir myfile <- ディレクトリの作成
vm1:/home/nakaj/Documents # ls -al
total 5
drwx----- 3 nakaj users  104 Oct 29 22:44 .
drwxr-xr-x  8 nakaj users  584 May 28 16:13 ..
-rw-r--r--  1 nakaj users 1119 May 28 16:13 .directory
drwxr-xr-x  2 root root    48 Oct 29 22:44 myfile

vm1:/home/nakaj/Documents # rmdir myfile <- ディレクトリの削除
vm1:/home/nakaj/Documents # ls -al
total 5
drwx----- 2 nakaj users   80 Oct 29 22:45 .
drwxr-xr-x  8 nakaj users  584 May 28 16:13 ..
-rw-r--r--  1 nakaj users 1119 May 28 16:13 .directory
```

## 名前の変更、移動(mv)

### ファイル名の変更 (mv)

```
vm1:/home/nakaj/Documents/myfile # ls -al
total 4
drwxr-xr-x 2 root root 72 Oct 29 22:34 .
drwx----- 4 nakaj users 152 Oct 29 17:31 ..
-rw-r--r-- 1 root root 143 Oct 29 22:34 my.txt
vm1:/home/nakaj/Documents/myfile #
vm1:/home/nakaj/Documents/myfile # mv my.txt your.txt <- ファイル名の変更
vm1:/home/nakaj/Documents/myfile # ls -al
total 4
drwxr-xr-x 2 root root 72 Oct 29 22:34 .
drwx----- 4 nakaj users 152 Oct 29 17:31 ..
-rw-r--r-- 1 root root 143 Oct 29 22:34 your.txt <-your.txtに変更された
vm1:/home/nakaj/Documents/myfile #
```

### ファイルを移動 (mv)

```
vm1:/home/nakaj/Documents/myfile # ls -al
total 4
drwxr-xr-x 2 root root 72 Oct 29 22:34 .
drwx----- 4 nakaj users 152 Oct 29 17:31 ..
-rw-r--r-- 1 root root 143 Oct 29 22:34 your.txt
vm1:/home/nakaj/Documents/myfile # mv your.txt ../backup <- 隣の backup に移動
vm1:/home/nakaj/Documents/myfile # ls -al
total 0
drwxr-xr-x 2 root root 48 Oct 29 22:38 .
drwx----- 4 nakaj users 152 Oct 29 17:31 ..
vm1:/home/nakaj/Documents/myfile # ls ../backup -al
total 4
drwxr-xr-x 2 root root 72 Oct 29 22:38 .
drwx----- 4 nakaj users 152 Oct 29 17:31 ..
-rw-r--r-- 1 root root 143 Oct 29 22:34 your.txt
```

## ファイルの属性、オーナーの変更 (chmod, chown)

### 属性の確認 (ls -l)

```
vm1:/home/nakaj/myfile # ls -al
total 4
drwxr-xr-x 2 root root 72 Oct 29 22:50 .
drwx----- 3 nakaj users 104 Oct 29 22:50 ..
-rw-r--r-- 1 nakaj users 143 Oct 29 22:50 my.txt
```

```
drwxr-xr-x 2 root root 72 Oct 29 22:50 .
drwx----- 3 maji users 104 Oct 29 22:50 ..
-rw-r--r-- 1 root root 143 Oct 29 22:50 my.txt
```

※-rw-r--r-- は [-][rwx][r--][r--] と分解して見ます。

[-]はファイル[d]はディレクトリです。

次の3桁は[オーナー][グループ][その他ユーザ]へのアクセス許可(Permission)を表します。  
通常8進数で表示し rwx=7(読み書き実行可) r--=4(読み込みのみ) rw-=6(読み書き可)です。

次の2カラムは[オーナー][グループ]です。

### 属性の変更 (chmod)

※ my.txt をオーナーが**実行(x)**できるようにします。

```
vm1:/home/nakaj/Documents/myfile # chmod 744 my.txt (744 オーナが実行可能)
vm1:/home/nakaj/Documents/myfile # ls -l
total 4
-rwxr--r-- 1 nakaj users 143 Oct 29 22:50 my.txt (x フラグが付きます)
```

※ □□□□□□□□□□□□□□□□□□□□□□□□ x(実行可)を付けないと実行できません。

### オーナーの変更 (chown)

※ my.txt を **wwwrun** ユーザ、**root** のグループに所有権を変更します。オーナーとグループの間は : (コロン) で区別します。

```
vm1:/home/nakaj/Documents/myfile # chown wwwrun:root my.txt
vm1:/home/nakaj/Documents/myfile # ls -l
total 4
-rwxr--r-- 1 wwwrun root 143 Oct 29 22:50 my.txt
```

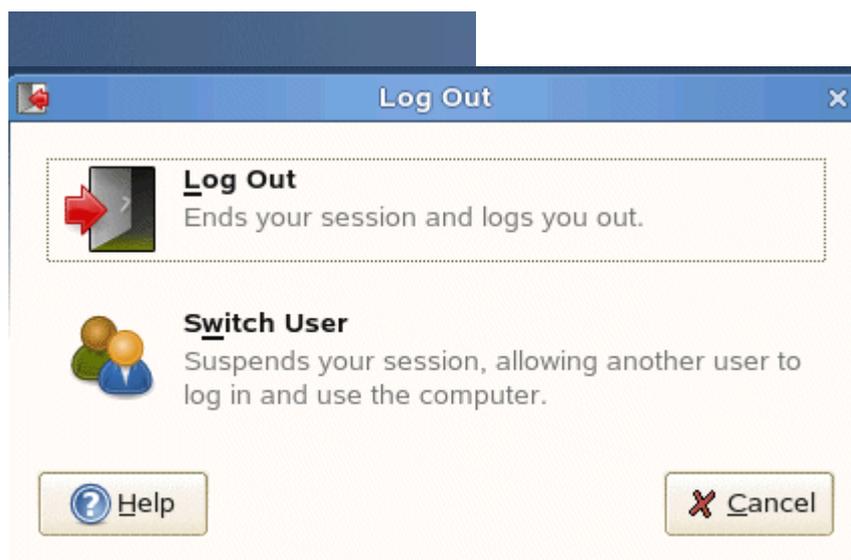
## X 端末の起動、終了、強制終了(startx の起動と終了)

### X 端末の起動

```
xenhost:~ # startx
```

### X 端末の終了

左端の Computer > Logout



Log Out をクリック

### X 端末の強制終了

コンソールで X 端末が制御不能になったときに実行します。X 上で動いているアプリケーションは全て強制終了します。

**Ctrl + Alt + BackSpace**

```
xenhost:~ # プロンプトに戻ります
```

## IP アドレスの確認と通信確認 ( ifconfig, ping )

### Ip アドレスの確認(ifconfig)

```
vm1:~ # ifconfig
eth0      Link encap:Ethernet  HWaddr 00:16:3E:44:ED:D6
          inet addr:192.168.1.190  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::216:3eff:fe44:edd6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:120 errors:0 dropped:0 overruns:0 frame:0
          TX packets:94 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:27137 (26.5 Kb)  TX bytes:11878 (11.5 Kb)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:37 errors:0 dropped:0 overruns:0 frame:0
          TX packets:37 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:4132 (4.0 Kb)  TX bytes:4132 (4.0 Kb)

vm1:~ #
```

### 通信確認(ping)

```
vm1:~ # ping 192.168.1.1 <-- 通信確認をします
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=250 time=0.458 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=250 time=0.306 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=250 time=0.327 ms <---Ctrl+Cで中止
--- 192.168.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 0.306/0.363/0.458/0.070 ms
vm1:~ #
```

## 通信経路の確認 (dig)

```

vm1:~ # dig @192.168.1.1 www.yahoo.co.jp (dig @自分の dns 相手のサーバ名)

; <<>> DiG 9.3.4 <<>> @192.168.1.1 www.yahoo.co.jp
; (1 server found)
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 56337
;; flags: qr rd ra; QUERY: 1, ANSWER: 13, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;www.yahoo.co.jp.                IN      A

;; ANSWER SECTION:
www.yahoo.co.jp.                139     IN      A      203.216.235.154
www.yahoo.co.jp.                139     IN      A      203.216.235.201
www.yahoo.co.jp.                139     IN      A      203.216.243.218
www.yahoo.co.jp.                139     IN      A      203.216.247.225
www.yahoo.co.jp.                139     IN      A      203.216.247.249
www.yahoo.co.jp.                139     IN      A      124.83.139.191
www.yahoo.co.jp.                139     IN      A      124.83.139.192
www.yahoo.co.jp.                139     IN      A      124.83.147.202
www.yahoo.co.jp.                139     IN      A      124.83.147.203
www.yahoo.co.jp.                139     IN      A      124.83.147.204
www.yahoo.co.jp.                139     IN      A      124.83.147.205
www.yahoo.co.jp.                139     IN      A      124.83.167.212
www.yahoo.co.jp.                139     IN      A      203.216.227.176

;; AUTHORITY SECTION:
yahoo.co.jp.                    129     IN      NS     dnsg01.yahoo.co.jp.
yahoo.co.jp.                    129     IN      NS     ns10.yahoo.co.jp.

;; ADDITIONAL SECTION:
ns10.yahoo.co.jp.               583     IN      A      210.80.243.9
dnsg01.yahoo.co.jp.            58245   IN      A      211.14.12.10

;; Query time: 367 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Thu Oct 23 16:53:41 2008
;; MSG SIZE rcvd: 313

```

vm1:~ #

※□□□ □□□□□□□□□□□□□□ **command | more** で画面スクロールを止めます。  
ex) # dig @mydns [www.yahoo.co.jp](http://www.yahoo.co.jp) | more

※□□□ nslookup も使用します

```

ex) nslookup www.novell.com
vm1:~ # nslookup www.novell.com
Server:          192.168.1.1 <- DNS サーバ
Address:         192.168.1.1#53

```

Non-authoritative answer:

```

Name:   www.novell.com
Address: 130.57.5.25 <- 相手先アドレス
vm1:~ #

```

## プロセスの管理 (top, ps, kill, free, sar)

## CPUを独占しているプログラムを調べて、強制終了させる(top)

```

vml:~ # top
top - 15:45:38 up 1 day, 3:19, 2 users, load average: 0.13, 0.06, 0.02
Tasks: 79 total, 1 running, 76 sleeping, 0 stopped, 2 zombie
Cpu(s): 0.0%us, 0.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 524452k total, 518092k used, 6360k free, 70068k buffers
Swap: 787144k total, 80k used, 787064k free, 267932k cached

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
12811	root	16	0	5524	1184	876	R	194	0.2	0:00.08	top
1	root	16	0	800	308	256	S	0	0.1	0:00.24	init
2	root	RT	0	0	0	0	S	0	0.0	0:01.97	migration/0
3	root	34	19	0	0	0	S	0	0.0	0:00.00	ksoftirqd/0
4	root	10	-5	0	0	0	S	0	0.0	0:00.00	events/0
5	root	10	-5	0	0	0	S	0	0.0	0:00.03	khelper
6	root	10	-5	0	0	0	S	0	0.0	0:00.06	kthread
7	root	10	-5	0	0	0	S	0	0.0	0:00.03	xenwatch
8	root	10	-5	0	0	0	S	0	0.0	0:00.00	xenbus
15	root	RT	-5	0	0	0	S	0	0.0	0:01.85	migration/1
16	root	34	19	0	0	0	S	0	0.0	0:00.00	ksoftirqd/1
17	root	10	-5	0	0	0	S	0	0.0	0:00.02	events/1
20	root	10	-5	0	0	0	S	0	0.0	0:00.00	kblockd/0
21	root	10	-5	0	0	0	S	0	0.0	0:00.00	kblockd/1
54	root	15	0	0	0	0	S	0	0.0	0:00.06	pdflush
56	root	15	0	0	0	0	S	0	0.0	0:01.25	kswapd0
57	root	11	-5	0	0	0	S	0	0.0	0:00.00	aio/0

どの COMMAND が %CPU の占有率が高いか、その PIDを確認します。

**k** を押して、次に PID をセット [Enter] PID を強制終了させます

```

top - 16:01:06 up 1 day, 3:35, 2 users, load average: 0.00, 0.00, 0.00
Tasks: 79 total, 1 running, 76 sleeping, 0 stopped, 2 zombie
Cpu(s): 0.0%us, 0.2%sy, 0.0%ni, 99.5%id, 0.3%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 524452k total, 507516k used, 16936k free, 70060k buffers
Swap: 787144k total, 80k used, 787064k free, 259620k cached

```

**Kill PID 57 with signal [15]: PID[enter]**

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
12883	root	16	0	5524	1180	872	R	0	0.2	0:00.17	top
1	root	16	0	800	308	256	S	0	0.1	0:00.24	init
2	root	RT	0	0	0	0	S	0	0.0	0:01.98	migration/0
3	root	34	19	0	0	0	S	0	0.0	0:00.00	ksoftirqd/0
4	root	10	-5	0	0	0	S	0	0.0	0:00.00	events/0
5	root	10	-5	0	0	0	S	0	0.0	0:00.03	khelper
6	root	10	-5	0	0	0	S	0	0.0	0:00.06	kthread
7	root	10	-5	0	0	0	S	0	0.0	0:00.03	xenwatch
8	root	10	-5	0	0	0	S	0	0.0	0:00.00	xenbus
15	root	RT	-5	0	0	0	S	0	0.0	0:01.85	migration/1
16	root	34	19	0	0	0	S	0	0.0	0:00.00	ksoftirqd/1
17	root	10	-5	0	0	0	S	0	0.0	0:00.02	events/1
20	root	10	-5	0	0	0	S	0	0.0	0:00.00	kblockd/0
21	root	10	-5	0	0	0	S	0	0.0	0:00.00	kblockd/1
54	root	15	0	0	0	0	S	0	0.0	0:00.06	pdflush
56	root	15	0	0	0	0	S	0	0.0	0:01.25	kswapd0
57	root	11	-5	0	0	0	S	0	0.0	0:00.00	aio/0

**q** で終了

## 不具合でハングアップしているプロセスの強制終了 (ps, kill)

※通常、こういうプロセスを”ゾンビ”と呼びます。例えば vi エディタを強制終了します。

```
vm1:~ # ps aux | grep vi
root      1446  0.0  0.6 13272 3252 ?          Ss   Oct27   0:00 /usr/sbin/hald
--daemon=yes --retain-privileges
root      9455  0.0  0.4  9628 2496 pts/3      S+   10:42   0:00 vi /etc/hosts
root      9457  0.0  0.1  3032  720 pts/1      S+   10:42   0:00 grep vi
vm1:~ #
```

← vi が /etc/hosts を開いている PID は 9455 であることを確認

```
vm1:~ # kill 9455
vm1:~ #
vm1:~ # ps aux | grep vi
root      1446  0.0  0.6 13272 3252 ?          Ss   Oct27   0:00 /usr/sbin/hald
--daemon=yes --retain-privileges
root      9459  0.0  0.1  3028  720 pts/1      S+   10:43   0:00 grep vi
vm1:~ #
```

← vi PID 9455 が停止した

※ヒント 画面に収まり切らない場合は `command | more` で画面スクロールを止めます。

ex) # ps aux | grep process\* | more

## - メモリの使用量を調べる free

abianca:~ # free パラメータなどブロック単位

	total	used	free	shared	buffers	cached
Mem:	510464	485704	24760	0	14320	256848
-/+ buffers/cache:		214536	295928			
Swap:	4193276	16632	4176644			

abianca:~ #

abianca:~ #

abianca:~ # free -m (MB 単位で調べる)

	total	used	free	shared	buffers	cached
Mem:	498	474	24	0	13	250
-/+ buffers/cache:		210	288			
Swap:	4094	16	4078			

abianca:~ # free -g (GN 単位で調べる)

	total	used	free	shared	buffers	cached
Mem:	0	0	0	0	0	0
-/+ buffers/cache:		0	0			
Swap:	3	0	3			

abianca:~ #

## – システムの CPU 利用状態を調べる sar

```
# sar
```

至近の10分おきの結果を表示します。sysstat パッケージをインストールすると利用できます。

```
abianca:~ # sar
```

```
Linux 3.0.13-0.27-xen (abianca)    08/01/12    _x86_64_
```

00:00:01	CPU	%user	%nice	%system	%iowait	%steal	%idle
00:10:01	all	0.19	0.00	0.13	0.44	0.19	99.04
00:20:01	all	0.20	0.00	0.14	0.39	0.19	99.09
00:30:01	all	0.19	0.00	0.14	0.38	0.19	99.10
00:40:01	all	0.19	0.00	0.13	0.35	0.19	99.14
00:50:01	all	0.19	0.00	0.13	0.42	0.19	99.07
01:00:01	all	0.20	0.00	0.13	0.58	0.19	98.90
01:10:01	all	0.19	0.00	0.13	0.38	0.19	99.11
01:20:01	all	0.19	0.00	0.13	0.37	0.19	99.11
01:30:01	all	0.19	0.00	0.13	0.34	0.19	99.15
01:40:01	all	0.19	0.00	0.12	0.34	0.19	99.15

```
#sar 1 -p 1秒おきにリアルタイムにチェックする。
```

```

10:31:41    all    0.00    0.00    0.25    1.24    0.25    98.26
10:31:42    all    0.25    0.00    0.51    1.52    0.25    97.47
10:31:43    all    0.00    0.00    0.50    1.00    0.50    98.00
10:31:44    all    0.00    0.00    1.01    1.01    0.00    97.98
10:31:45    all    0.00    0.00    0.00    1.01    0.51    98.48
10:31:46    all    0.25    0.00    0.76    0.76    0.25    97.98
10:31:47    all    0.00    0.00    0.77    1.54    0.26    97.43
10:31:48    all    0.00    0.00    0.50    12.06    0.25    87.19
10:31:49    all    0.00    0.00    1.01    12.91    0.25    85.82
10:31:50    all    0.00    0.00    0.51    1.02    0.00    98.47
10:31:51    all    0.00    0.00    0.51    1.52    0.51    97.47
10:31:52    all    0.25    0.00    1.24    1.74    0.25    96.52
10:31:53    all    0.00    0.00    1.01    1.26    0.25    97.49
10:31:54    all    0.25    0.00    1.00    10.00    0.25    88.50
10:31:55    all    0.00    0.00    0.25    2.27    0.25    97.22
10:31:56    all    0.00    0.00    0.76    0.76    0.00    98.49
10:31:57    all    0.00    0.00    0.76    3.04    0.25    95.95

10:31:57    CPU    %user    %nice    %system    %iowait    %steal    %idle
10:31:58    all    0.00    0.00    0.50    0.75    0.25    98.50
10:31:59    all    0.00    0.00    0.25    1.52    0.25    97.98
10:32:00    all    0.00    0.00    0.50    1.00    0.25    98.25
10:32:01    all    0.00    0.00    0.25    6.77    0.50    92.48

```

%iowait が多い場合、IO の待ちが多く発生しています。%steal が多い場合は仮想マシン同士の CPU リソースの奪い合いでパフォーマンスが落ちています。

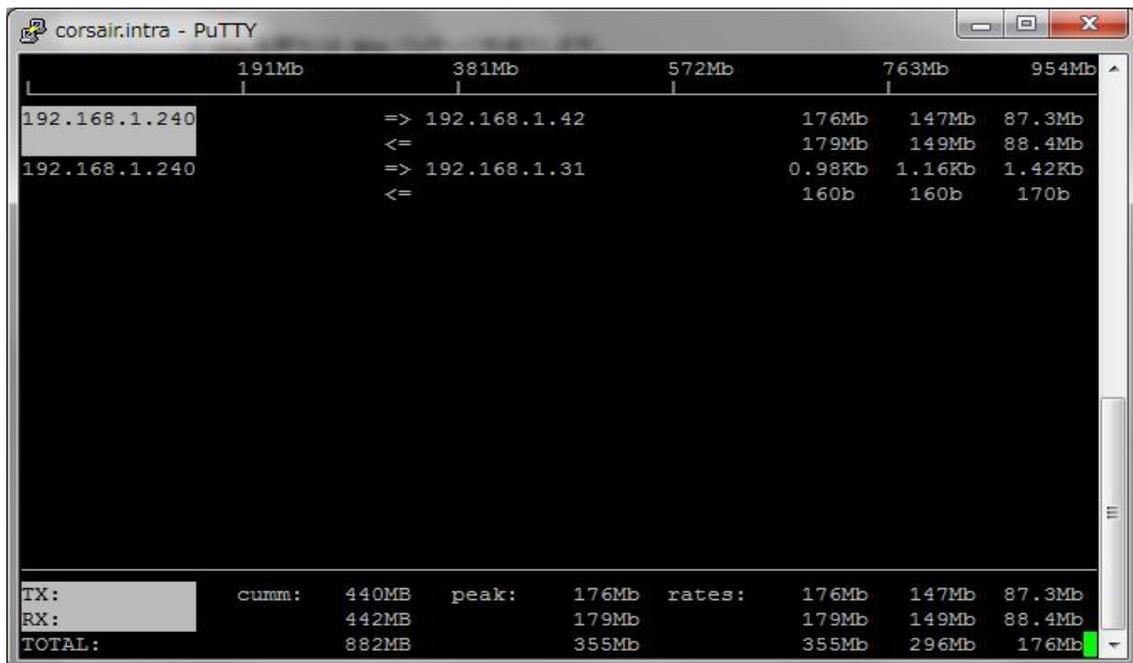
sar の代わりに # mpstat 1 も利用できます。

- 通信量をリアルタイムにチェックする iftop

※ iftop を使うには iftop パッケージを導入します。

```
# iftop -i [インターフェース名]
```

インターフェース名は eth0 から br0 などの名前です。Ifconfig コマンドで確認してください。白いグラフの部分は転送中の状態を示します。

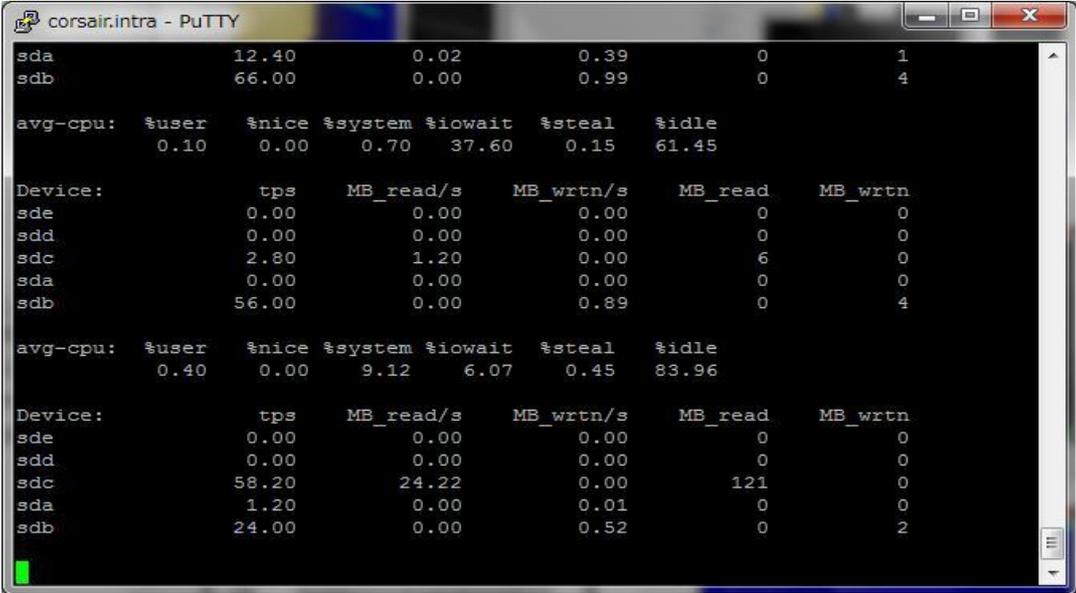


q で終了します。

- 転送中のディスクの状態を調べたい iostat

```
# iostat -hm [秒]
```

-hm は(Human Readable で MB 表記を示します。Kb 表記にしたい場合は -hk オプションを使います。[秒]は更新間隔を示します。更新間隔を1秒に設定すると情報はリアルタイムですが、平均的な速度を見たい場合は 5 ~ 10 秒程度を設定すると良いでしょう。



```
corsair.intra - PuTTY
sda      12.40    0.02    0.39     0      1
sdb      66.00    0.00    0.99     0      4

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.10    0.00    0.70   37.60    0.15   61.45

Device:            tps    MB_read/s    MB_wrtn/s    MB_read    MB_wrtn
sde                 0.00         0.00         0.00         0          0
sdd                 0.00         0.00         0.00         0          0
sdc                 2.80         1.20         0.00         6          0
sda                 0.00         0.00         0.00         0          0
sdb                 56.00         0.00         0.89         0          4

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.40    0.00    9.12   6.07    0.45   83.96

Device:            tps    MB_read/s    MB_wrtn/s    MB_read    MB_wrtn
sde                 0.00         0.00         0.00         0          0
sdd                 0.00         0.00         0.00         0          0
sdc                 58.20        24.22         0.00        121         0
sda                 1.20         0.00         0.01         0          0
sdb                 24.00         0.00         0.52         0          2
```

tps は転送速度です。

## ファイルの内容を確認する (cat, more, less, tail)

### 先頭から表示する(cat) - 短いファイル向け

```
vm1:~/Desktop # cat /var/log/messages
May 28 07:09:05 linux syslog-ng[2118]: syslog-ng version 1.6.8 starting
May 28 07:09:09 linux ifup: Service network not started and mode 'auto' -> skipping
May 28 07:09:10 linux kernel: klogd 1.4.1, log source = /proc/kmsg started.
May 28 07:09:10 linux kernel: AppArmor: AppArmor (version 2.0-19.43r6320) initialized
```

以下略

### 先頭からスクロールさせながら表示する (more) - 長いファイル向け

```
vm1:~/Desktop # more /var/log/messages
May 28 07:09:05 linux syslog-ng[2118]: syslog-ng version 1.6.8 starting
May 28 07:09:09 linux ifup: Service network not started and mode 'auto' -> skipping
```

space キーで次のページへ q で終了

### 先頭からスクロールさせながら表示する (less) - 上下にスクロールできる

```
vm1:~/Desktop # less /var/log/messages
May 28 07:09:05 linux syslog-ng[2118]: syslog-ng version 1.6.8 starting
May 28 07:09:09 linux ifup: Service network not started and mode 'auto' -> skipping
```

space キーで次のページへ、カーソルキーでスクロール、q で終了

### 最後のページだけ表示する(tail)

```
vm1:~/Desktop # tail /var/log/messages
: 略
Oct 29 14:46:33 vm1 sshd[24364]: Accepted keyboard-interactive/pam for root from 192.168.1.11 port 1542 ssh2
Oct 29 14:50:59 vm1 sshd[25241]: Accepted keyboard-interactive/pam for root from 192.168.1.11 port 1543 ssh2
Oct 29 15:07:37 vm1 sshd[25397]: Accepted keyboard-interactive/pam for root from 192.168.1.11 port 1550 ssh2
vm1:~/Desktop #
```

## ディスクの容量を確認する (df, du)

df ディスク全体の容量を確認する

```
xenhost:~ # df -l
Filesystem      1K-blocks      Used Available Use% Mounted on
/dev/sda4        118250848 109852664   8398184   93% /
udev             1860624      216   1860408    1% /dev
/dev/sda5        88081656   76592336  11489320   87% /backupPC
xenhost:~ #
```

```
xenhost:~ # df -h (-h オプションで Gb 単位で表示されます)
```

```
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda2        1.8T  1.3T  425G   76% /
devtmpfs         7.9G  444K   7.9G    1% /dev
tmpfs            7.9G  6.5M   7.9G    1% /dev/shm
/dev/sdc1        1.8T  1.6T  132G   93% /disk2
/dev/sdd2        1.8T  1.4T  335G   81% /disk3
/dev/sdb3        1.4T  1.4T   40G   98% /disk1
xenhost:~ #
```

du 特定のディレクトリのファイル使用量を確認する

```
xenhost:~ # du -h -c /var/lib/xen
17G    /var/lib/xen/images/zemen
17G    /var/lib/xen/images/proxy
132G   /var/lib/xen/images/eilian
22G    /var/lib/xen/images/vampire
7.8G   /var/lib/xen/images/zepher
31G    /var/lib/xen/images/w732
14G    /var/lib/xen/images/zcm111
21G    /var/lib/xen/images/dns3
7.0G   /var/lib/xen/images/w832
29G    /var/lib/xen/images/mcat
9.1G   /var/lib/xen/images/zens
9.8G   /var/lib/xen/images/w8x64
```

-c オプションを付けると最後に .Total が表示されます。

Linux の EXT3 フォーマットは Windows NTFS のようにフラグメンテーションが発生しにくいシステムですが 空き容量が10%を切るとフラグメンテーションが発生する場合があります。

## パスワードの変更(passwd、smbpasswd)

### ログオンパスワードの変更(passwd)

※通常簡単なパスワード(数字のみ、ユーザ名と一致、短すぎ)を設定すると警告されますが、変更は受け付けられます。なるべく英数字を組み合わせた6-8文字のパスワードを設定します。

```
vm1:~ # passwd nakaj <- ユーザ nakaj のパスワードを変更
Changing password for nakaj.
New Password:簡単なパスワード
Bad password: too simple
Reenter New Password:
Passwords do not match. <- Reenter が違う
New Password:複雑なパスワード
Reenter New Password: Reenter する
Password changed.<- パスワードが変更された
vm1:~ #
```

### samba ユーザの作成とパスワードの変更(smbpasswd)

※ samba のパスワードは通常のログインIDとは別に管理されています。Samba をインストールした後、必ず `smbpasswd -a` コマンドでユーザの作成とパスワードを設定します。

```
vm1:~ # smbpasswd -a nakaj <- -a オプションでユーザを追加(add)します
New SMB password: <- パスワードを設定
Retype new SMB password:
Added user nakaj.

vm1:~ #
vm1:~ # smbpasswd nakaj
New SMB password: <- パスワードを変更
Retype new SMB password:
vm1:~ #
```

## シェルを実行する (バッチジョブ)

### シェル(バッチジョブ)を実行

シェルはテキストエディタで作成したあと、シェルファイルに x (実行フラグ)を付けると実行可能になります。

```
vm1:/home/nakaj/Documents # ls -al hello.sh <- 属性を確認
-rw-r--r-- 1 root root 20 Oct 28 11:59 hello.sh <- まだ x フラグがない
```

```
vm1:/home/nakaj/Documents # cat hello.sh <- テキストとして読める
echo Hello World ! <- "Hello World !"をスクリーンにエコーバック表示
```

```
vm1:/home/nakaj/Documents # ./hello.sh <- x がないまま実効
-bash: ./hello.sh: Permission denied <- 実行できない
```

```
vm1:/home/nakaj/Documents # chmod 744 hello.sh <- rwx の属性を与える
vm1:/home/nakaj/Documents # ls -al hello.sh
-rwxr--r-- 1 root root 20 Oct 28 11:59 hello.sh <- 属性が与えられた
```

```
vm1:/home/nakaj/Documents # ./hello.sh <- 実行する
Hello World ! <- 実行できた
```

※ シェルを含むコマンドプログラムは絶対パス( /home/myname/shell.sh )で実行するか、現在のディレクトリにある場合は ./ を先頭に付けます。

## XM 仮想マシンの起動とシャットダウン、強制終了(xm list, shutdown, destroy, create)

```

xenhost:~ # xm list
Name                    ID    Mem VCPUs      State    Time(s)
Domain-0                0    679    2    r----- 94309.0
W2K3S                   1    512    1    r----- 690769.3
vm1                    7    512    2    -b----- 45.4
oes2ltest               1250  2      2      90053.3

xenhost:~ # xm shutdown 7 (ID:7 vm1 を遮断)
xenhost:~ #
xenhost:~ # xm list
Name                    ID    Mem VCPUs      State    Time(s)
Domain-0                0    679    2    r----- 94313.9
W2K3S                   1    512    1    r----- 690810.6
vm1                    512    2      53.6
oes2ltest               1250  2      2      90053.3

xenhost:~ # (ID:7 がないことを確認)
xenhost:~ # xm create vm1 (vm1 を create-開始)
Using config file "/etc/xen/vm/vm1".
Started domain vm1
xenhost:~ #
xenhost:~ # xm list
Name                    ID    Mem VCPUs      State    Time(s)
Domain-0                0    679    2    r----- 94321.5
W2K3S                   1    512    1    r----- 690848.4
vm1                    8    512    2    -b----- 20.2
oes2ltest               1250  2      2      90053.3

xenhost:~ # (ID:8 で起動したことを確認)
xenhost:~ # xm destroy 8 (Destroy-強制終了)
xenhost:~ #
xenhost:~ # xm list
Name                    ID    Mem VCPUs      State    Time(s)
Domain-0                0    679    2    r----- 94327.9
W2K3S                   1    512    1    r----- 691271.9
vm1                    512    2      20.2
oes2ltest               1250  2      2      90053.3

xenhost:~ # (ID:8 がないことを確認)

```

## 仮想コンソールの操作( xm list, xm console)

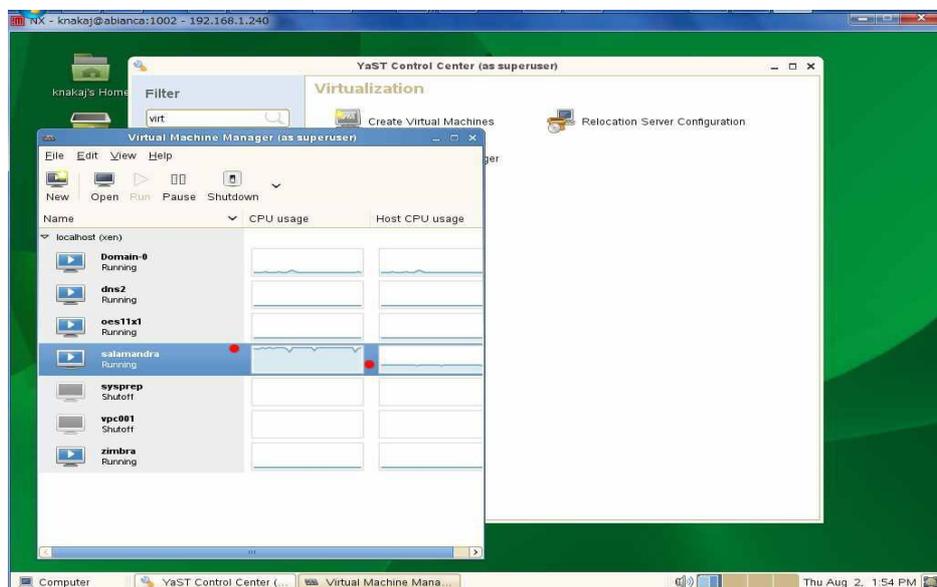
```
xenhost:~ # 親サーバのプロンプト
xenhost:~ # xm list
Name           ID   Mem VCPUs   State   Time(s)
Domain-0       0   679    2   r----- 94300.8
W2K3S          1   512    1   r----- 689835.7
vm1           7   512    2   -b----- 41.7
oes2ltest      1250 2     2     90053.3
xenhost:~ # 子サーバ vm1 (ID:7) にコンソールを切り替えます
xenhost:~ # xm console 7
[enter キー]

Welcome to SUSE Linux Enterprise Server 10 SP2 (x86_64) - Kernel 2.6.16.60-0.21-
xen (xvc0).

vm1 login: root (子サーバのプロンプト、ログインします)
Password:*****
Last login: Thu Oct 23 16:03:44 JST 2008 on xvc0
vm1:~ #
vm1:~ #
vm1:~ #
vm1:~ # xenhost:~ # (Ctrl+] を押す)
xenhost:~ #
xenhost:~ # プロンプトが親に戻ります
```

## 仮想マシンがハングアップした場合

仮想マシンがハングアップしたり、ブルースクリーン状態に陥ると、Virt-Manager からは、一定のタスクが走っており、CPUも動作しているように見えます。ただしCPUの利用率は一定です。



このような場合、仮想マシンを完全に destroy して create します。

```
# xm destroy crashed-VM
```

```
# xm create crashed-VM
```

```
mydom0:~ # xm create vpc001
Using config file "/etc/xen/vm/vpc001".
Started domain vpc001 (id=9) <- VM が起動しました。
mydom0:~ #
```

VM が起動したら、右ボタンから open して起動状態を確認してください。ローカルログインして ping などの通信系を確認します。

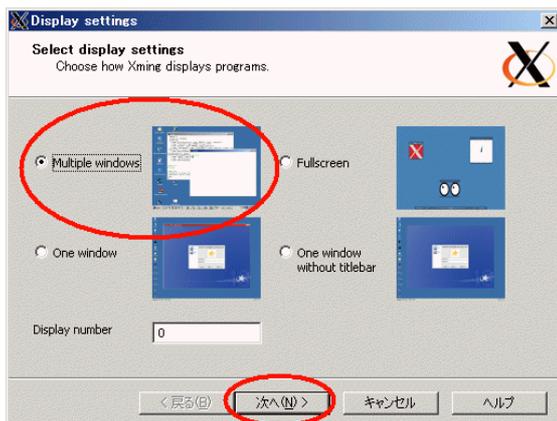
## Windows から仮想 X 端末を使う(xming)

Windows マシンから GUI のソフトウェアを利用します

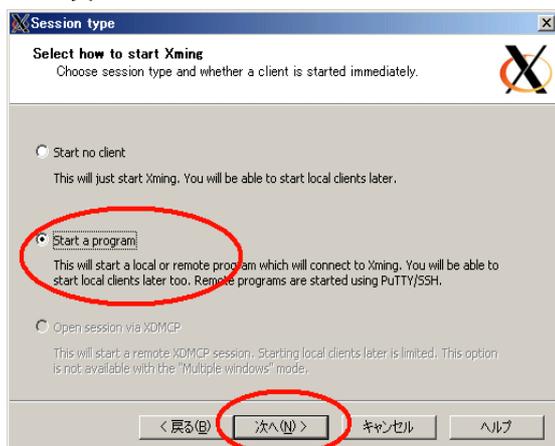
xming/Xlaunch のアイコンをクリック



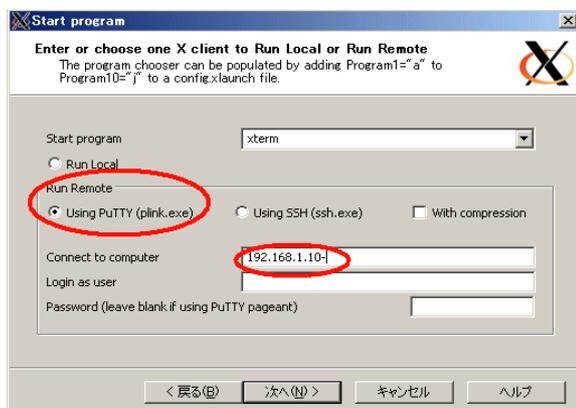
Multiple Windows > 次へ



Start program > 次へ

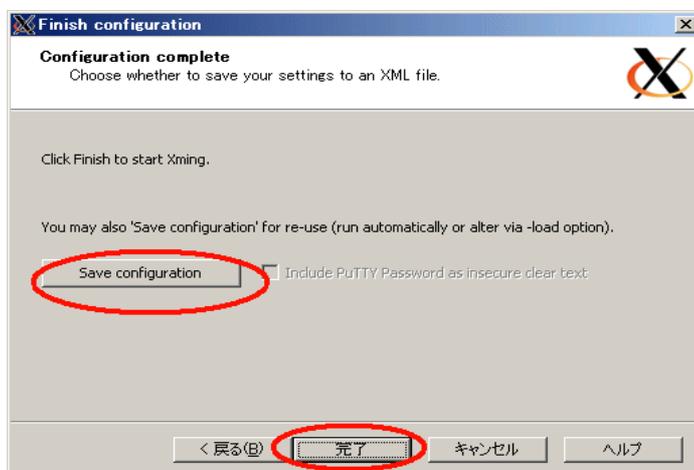


Using Putty(plink.exe) と Connect to Computer にコンピュータ名、またはアドレスをセットして次へ



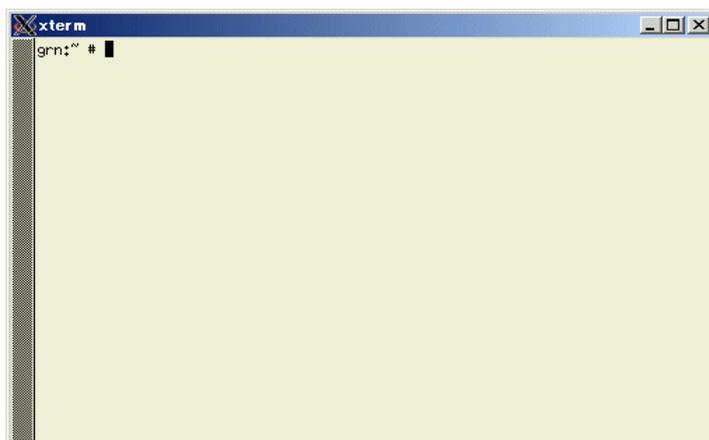
## そのまま完了

※ または、Save Configuration を選んで、判りやすい名前(MyComputer.config) を付けて、デスクトップやプログラムフォルダに保存します。



## ログイン

ログインダイアログ(ログイン名) OK > パスワード(パスワード) OK の小さなダイアログが出ます。



## xterm から GUI プログラムの起動

```
mypc:~# yast2 &           - yast(コンピュータの設定)
mypc:~# nautilus &        - nautilus(ファイルブラウザ)
mypc:~# firefox &         - firefox(インターネットブラウザ)
mypc:~# soffice &         - soffice(オフィスソフトウェア - インストールされている場合のみ)
mypc:~# xterm &           - xterm(もうひとつ端末を開く)
mypc:~# gedit &           - gedit(テキストエディタを開く)
```

※ コマンド名の後に & を付けるとバックグラウンドジョブ(端末を占有しない)で起動できます。  
※ これらのプログラムは Windows 上で稼動しているわけではありません。Linux 上で動作しています。コピーペーストの動作に一部問題がある場合があります。

## 終了

タスクトレイの xming アイコンを右クリックして exit します。



## xterm から GUI プログラムの起動

```
mypc:~# yast2 &           - yast(コンピュータの設定)
mypc:~# nautilus &        - nautilus(ファイルブラウザ)
mypc:~# firefox &        - firefox(インターネットブラウザ)
mypc:~# soffice &         - soffice(オフィスソフトウェア - インストールされている場合のみ)
mypc:~# xterm &           - xterm(もうひとつ端末を開く)
mypc:~# gedit &           - gedit(テキストエディタを開く)
```

※ コマンド名の後に & を付けるとバックグラウンドジョブ(端末を占有しない)で起動できます。  
※ これらのプログラムは Windows 上で稼動しているわけではありません。Linux 上で動作しています。結果が画面に映し出されているだけです。コピーペーストの動作に一部問題がある場合があります。

## 終了

タスクトレイの xming アイコンを右クリックして exit します。



## ファイルの圧縮と解凍( tar )

テクニカルサポートを受けるため、ログファイルを圧縮してひとつのファイルにまとめます。  
/var/log 以下のシステムログファイルを圧縮して1つのファイルにまとめる例です。  
はじめに cd コマンドで /var/log に移動します。  
tar コマンドで、ユーザのホームディレクトリに logfile.tar.gz というファイルにまとめます。

```
vm1:~ # cd /var/log
vm1:/var/log # tar -cvzf /home/nakaj/logfiles.tar.gz .
./
./r
./gdm/
./ntp
./cups/
./cups/access_log
./cups/error_log
./krb5/
./mail
./news/
./news/news.err
./news/news.notice
./news/news.crit
: 略
```

※ **-cvzf** オプションは **C**reate(作成) **V**erboze(詳細表示しながら) **Z**ip(圧縮) **F**ile 名の略です  
※ **-cvzf** オプションで圧縮したファイルは、通常 **xxxx.tar.gz** という拡張子を指定します。  
※ **-f** オプションが付いているので必ずディレクトリを指定します。  
ここでは **.** (ドット) カレントディレクトリ以下すべてです。

## 圧縮内容の確認と解凍 ( -tvzvf と -xvzvf オプション)

### 内容確認 (-tvzvf)

```

vml:/home/nakaj/Documents/myfile # tar -tvzvf logfile.tar.gz
drwxr-xr-x root/root          0 2008-10-30 16:02:39 ./
-rw-r--r-- root/root    2018861 2008-10-30 16:07:39 ./r
drwxr-x--- gdm/gdm          0 2008-04-24 17:45:42 ./gdm/
-rw-r--r-- root/root          0 2008-05-28 16:02:35 ./ntp
drwxr-xr-x lp/lp            0 2008-10-29 23:11:30 ./cups/
: 以下略

```

※ -tvzvf オプションは圧縮された tar.gz 形式のファイルの内容をリストします。正しく圧縮されたか確認することができます。

### 解凍(-xvzvf)

```

vml:/home/nakaj/Documents/myfile # tar -xvzvf logfile.tar.gz
./
./r
./gdm/
./ntp
./cups/
./cups/access_log
./cups/error_log
./krb5/
./mail
./news/
./news/news.err
./news/news.notice
: 略

```

※ -xvzvf オプションは圧縮された tar.gz 形式のファイルの内容を解凍 (eXtract) します

```

vml:/home/system/Documents/myfile # ls -al <-解凍されたか確認
total 7331
drwxr-xr-x 12 root  root    1400 Oct 30 16:02 .
drwx----- 3 maji  users    104 Oct 30 16:12 ..
-rw-r--r--  1 root  root   4557 May 28 16:52 SaX.log
lrwxrwxrwx  1 root  root     10 Oct 30 16:14 XFree86.0.log ->
Xorg.0.log
lrwxrwxrwx  1 root  root     14 Oct 30 16:14 XFree86.0.log.old ->
Xorg.0.log.old
-rw-r--r--  1 root  root  13624 Jun 17 17:54 Xorg.0.log
-rw-r--r--  1 root  root  17320 May 29 15:03 Xorg.0.log.old
drwx-----  2 root  root     640 May 28 16:50 YaST2
: 略

```

ヒント : ファイルはエクスプローラから見えるけど、読めない、書けない！

圧縮したログファイルのオーナーを変える (chown)

※せっかく圧縮して samba で取り出そうとしても、root が作成したファイルであるため、nakaj でログインした samba からは読めても書けない、名前の変更ができない場合があります。

chown でオーナーを変更します。

```
vm1:/home/nakaj/Documents/myfile # ls -al
total 3051
drwxr-xr-x 2 root    root      80 Oct 30 16:33 .
drwx----- 3 root    users     136 Oct 30 16:32 ..
-rw-r--r-- 1 root    root    3120362 Oct 30 16:33 logfiles.tar.gz
vm1:/home/nakaj/Documents/myfile #
vm1:/home/nakaj/Documents/myfile # chown nakaj:users logfiles.tar.gz
vm1:/home/nakaj/Documents/myfile # ls -al
total 3051
drwxr-xr-x 2 root    root      80 Oct 30 16:33 .
drwx----- 3 root    users     136 Oct 30 16:32 ..
-rw-r--r-- 1 nakaj  users   3120362 Oct 30 16:33 logfiles.tar.gz
vm1:/home/nakaj/Documents/myfile #
```

※ ファイルだけではなく /home/nakaj/ 以下に作成されたサブディレクトリが root のオーナーになっている場合は、書き込みはできません。この例では .... /myfile も chown しなければ、Windows からファイルの書き込みはできません。

## rsync を使った Linux 間ファイルコピー

rsync を使った Linux → Linux 間のファイルコピー、バックアップの方法です。コピー元から次のようにコマンドを使います。

```
Linux:~ # rsync -avz -e ssh /root/Desktop/ 192.168.1.240:/root/Documents/Backup

The authenticity of host '192.168.1.240 (192.168.1.240)' can't be established.
RSA key fingerprint is 00:9c:a5:d4:21:b6:96:3e:5f:64:e8:31:99:ab:3c:3e.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
Warning: Permanently added '192.168.1.240' (RSA) to the list of known hosts.
Password:*****
building file list ... done
created directory /root/Documents/Backup
./
myfile/
myfile/filename-1.txt
myfile/filename-2.txt
myfile/filename-3.txt

: 中略

sent 130671051 bytes  received 200152 bytes  1756660.44 bytes/sec
total size is 170789004  speedup is 1.31
Linux:~ #
```

※ コピー元で mydire/ と最後に / を付けた場合 “mydir/以下のファイル” 全て mydir と / を省略した場合は “mydir ディレクトリごと” コピーします。

※ --delete オプションを付けるとコピー元から削除されたファイルはコピー先からも削除されます(同期)

## scp を使ったファイルコピー

root ユーザが IwantCopyThisFile を destination サーバの /root ディレクトリに SSH コピーするには

```
# scp IwantCopyThisFile root@distination:/root
```

を実行します

相手の root パスワードを聞いてくるのでパスワードをセットします。

※ コピーするファイルが仮想マシンのような大きなサイズで時間がかかる場合、**-c arcfour** オプションで暗号化レベルを下げることで転送速度が上がる場合があります。ただしセキュリティレベルは低くなるため、安全な環境で使用してください。

```
source:/var/lib/xen/images/proxy # scp proxy.disk0.raw root@distination:/root
root@distination's password:
proxy.disk0.raw                                100%   16GB   51.7MB/s   05:17

source:/var/lib/xen/images/proxy # scp -c arcfour proxy.disk0.raw
root@distination:/root
root@distination's password:
proxy.disk0.raw                                100%   16GB   61.1MB/s   04:28
source:/var/lib/xen/images/proxy #
```

## ※SSH 接続全般の注意

初回 ssh 接続する場合、相手の発行するキーを受け入れます。キーは /home/myhome/.ssh/known\_hosts にあ

ります。

```
myhost:~ # rm .ssh/known_hosts <--- キーファイルを削除します。
myhost:~ #
myhost:~ # ssh 192.168.1.112 <---- destination に ssh 接続します。
```

```
The authenticity of host '192.168.1.112 (192.168.1.112)' can't be established.
RSA key fingerprint is cc:c9:a5:50:dd:a4:f2:ba:c4:b2:1b:b4:c3:7a:f0:b5.
Are you sure you want to continue connecting (yes/no)? yes <--キーを受け入れる
Warning: Permanently added '192.168.1.112' (RSA) to the list of known hosts.
```

```
Password: ***** <---- 相手のパスワード
Last login: Thu Mar  8 09:03:21 2012 from 192.168.1.35
destination:~ #
```

※ Windows の場合は次のキーの下に受け入れ済みキーが格納されています。この下のキーが known\_hosts の役割を果たします。

```
[HKEY_CURRENT_USER\Software\SimonTatham\PuTTY\SshHostKeys]
```

## ハードウェア管理ツール

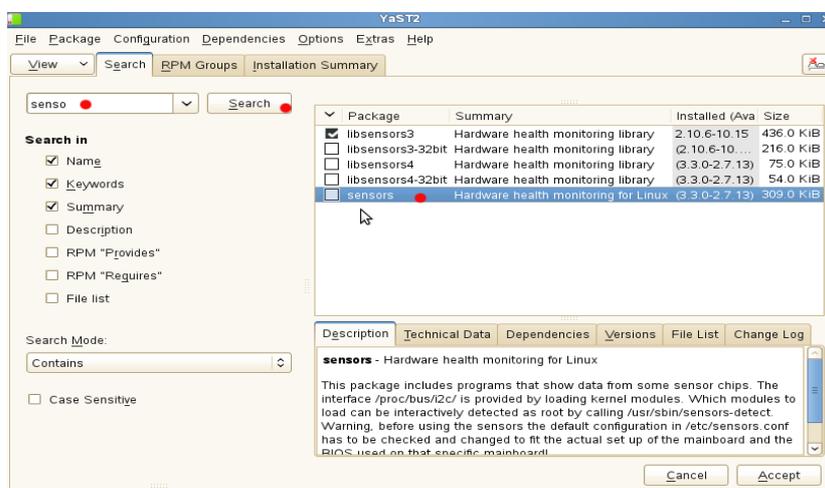
デフォルトでインストールされない、ハードウェアの管理ツールを導入します。

※ これらの機能はインストール後再起動が必要な場合があります。本格的な運用前に設定して確認します。

YaST > Software Management から次の管理ツールをインストールします。

- 1) smartctltools
- 2) sysstat
- 3) sensors
- 4) pmtools

Search ボックスから上のパッケージを Search してチェックボックスをチェックして Accept ボタンを押してインストールします。



## 温度センサー sensors

# sensors-detect コマンドを実行し再起動します

※ sensors は detect 後、再起動が必要な場合があります。また機器、Linux のカーネルバージョンによっては正常に検出できない場合があります。必ず最新の sensors パッケージを使用してください。

```
sles:~ # sensors-detect
# sensors-detect revision 5337 (2008-09-19 17:05:28 +0200)
```

```
This program will help you determine which kernel modules you need
to load to use lm_sensors most effectively. It is generally safe
and recommended to accept the default answers to all questions,
unless you know what you're doing.
```

```
We can start with probing for (PCI) I2C or SMBus adapters.
Do you want to probe now? (YES/no): y または[ENTER]
```

この間のインタビューはすべてデフォルト(Yes)で答えます。適合したチップセットのセンサーを検出します。

```
* Chip `AMD K10 thermal sensors' (confidence: 9)
```

```
Do you want to overwrite /etc/sysconfig/lm_sensors? (YES/no): y または[ENTER]
sles:~ #
```

センサーが認識できた場合、再起動後、sensors コマンドで CPU 周辺温度の監視ができるようになります。

```
sles:~ # sensors
radeon-pci-0008
Adapter: PCI adapter
temp1: +48.0°C

k10temp-pci-00c3
Adapter: PCI adapter
temp1: +48.1°C (high = +70.0°C)
(crit = +75.0°C, hyst = +72.0°C)
```

## smartctl で S.M.R.T 情報を確認

```
sles:~ # smartctl -a /dev/sda (デバイス名は機器により異なります)
smartctl 5.39 2008-10-24 22:33 [x86_64-suse-linux-gnu] (openSUSE RPM)
Copyright (C) 2002-8 by Bruce Allen, http://smartmontools.sourceforge.net

=== START OF INFORMATION SECTION ===
Device Model:          Hitachi HDS722020ALA330
Serial Number:        JK1131YAGZJKXV
Firmware Version:    JKAOA20N
User Capacity:        2,000,398,934,016 bytes
Device is:            Not in smartctl database [for details use: -P showall]
ATA Version is:      8
ATA Standard is:     ATA-8-ACS revision 4
Local Time is:       Fri Apr 13 10:35:37 2012 JST
SMART support is:    Available - device has SMART capability.
SMART support is:    Enabled
```

以下省略

## dmidecode ハードウェア情報の取得

```
# dmidecode

Thu Apr 12 20:05:26 JST 2012
# dmidecode 2.9
SMBIOS 2.6 present.
53 structures occupying 1859 bytes.
Table at 0x0009F800.

Handle 0x0000, DMI type 0, 24 bytes
BIOS Information
    Vendor: American Megatrends Inc.
    Version: V11.7
    Release Date: 10/22/2010
    Address: 0xF0000
    Runtime Size: 64 kB
    ROM Size: 1024 kB
```

- 以下略 -

## 免責事項

この文書はすべての条件で動作を保障したものではありません。また、この文書によって発生したいかなる障害、損害についてのお問い合わせはお受けしません。

この文書に関するお問い合わせは、当所との業務契約中のお客様に限りお受けします。あらかじめご了承の上ご参考ください。

この文書はあくまでも参考として公開しています。正規のサポートはノベル株式会社のプレミアムサービス付きアクティベーションキーをご購入の上ご利用ください。

この文書は予告なく変更されます。

各社の商品名、商標は各社に所属します。