

Pasos para la instalación de PVM

Para que funcione PVM es necesario

1. Que rsh (remote shell) esté funcionando en todas las máquinas a utilizarse, esto es necesario para que cada hosts pueda ejecutar programas en hosts remotos.
2. Verificar que en el archivo /etc/hosts no exista un alias a 127.0.0.1
3. Instalar PVM
4. Ejecutar PVM

Pasos a seguir para la instalación en el laboratorio del Departamento de Ciencias e Ingeniería de la Computación:

0. loggarse como user: pcs

1. Configuración de rsh y rlogin

Ejecutar el comando "su" y luego poner el password de root

```
$su
password:
#
```

a) EDITAR el archivo /etc/xinetd.d/rsh y poner disable= no

```
service shell
{
    socket_type           = stream
    wait                 = no
    user                 = root
    log_on_success       += USERID
    log_on_failure       += USERID
    server               = /usr/sbin/in.rshd
    disable              = no
}
```

b) EDITAR el archivo /etc/xinetd.d/rlogin y poner disable= no

```
service login
{
    socket_type           = stream
    wait                 = no
    user                 = root
    log_on_success       += USERID
    log_on_failure       += USERID
    server               = /usr/sbin/in.rlogind
    disable              = no
}
```

c) Reiniciar el demonio xinetd

```
[root@pc09 xinetd.d]# cd ..
[root@pc09 etc]# cd init.d/
[root@pc09 init.d]# ./xinetd restart
Stopping xinetd:           [ OK ]
Starting xinetd:          [ OK ]
[root@pc09 init.d]#
```

d) Indicar que otros hosts remotos pueden ejecutar en esta máquina

d.1 pasar a modo usuario pcs, (simplemente hacer exit)

```
# exit
$
```

d.2 Como usuario pcs, crear/modificar el archivo `.rhosts` y agregar una línea por cada host que pueda acceder indicando `<host.dominio> <usuario>`

Por ejemplo, en el laboratorio del subsuelo sería:

```
Pc04.clab1.cs.uns.edu.ar    pcs
Pc05.clab1.cs.uns.edu.ar    pcs
localhost                   pcs
```

d.3) el archivo `.rhosts` tiene que quedar con los permisos `-rw-----`
se cambia con

```
$ chmod 600 .rhosts
usar ls -al para ver los permisos
```

e) Para verificar desde una máquina distinta a la dos

```
$ rsh Pc02
-(...)$
```

2. Modificar el archivo `/etc/hosts`, esto debe hacerse como root

```
$ su
Password
# pico /etc/hosts
```

comentar la línea que contiene la dirección de loopback:

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
#127.0.0.1          Pc09.clab1.cs.uns.edu.ar  Pc09
192.168.0.20       bastet.cs.uns.edu.ar  bastet
```

3 Instalar pvm (seguir como root)

a.1) descomprimir el tar.gz, ubicado por ejemplo en /usr/local o usr/share

```
[root@pc09 share]# gunzip pvm3.4.4.tar.gz
[root@pc09 share]# tar xvf pvm3.4.4.tar
```

a.2) definir la variable de entorno

```
[root@pc09 etc]# PVM_ROOT=/usr/share/pvm3
[root@pc09 etc]# export PVM_ROOT
```

a.3) hacer make del pvm

```
[root@pc09 pvm3]# cd /usr/share/pvm3
[root@pc09 pvm3]# make
```

a.4) modificar los permisos del directorio LINUX

```
[root@pc09 pvm3]# cd /usr/share/pvm3/bin
[root@pc09 pvm3]# chmod 757 LINUXI386/
```

b) crear/editar .bashrc en la cuenta local de pcs (salir de root)

```
[root@pc09 lib]# exit
bash-2.05b$ cd
```

b.1) verificar si existe el .bashrc

```
bash-2.05b$ ls -la
total 16
drwxr-xr-x  2 pcs  users  4096 Oct 28 17:24 .
drwxr-xr-x  6 root root   4096 Oct 28 17:18 ..
-rw-----  1 pcs  users   81 Oct 28 17:25
.bash_history
-rw-----  1 pcs  users   99 Oct 28 17:23 .rhosts
```

b.2) copia y genera el .bashrc

```
bash-2.05b$ cat /usr/share/pvm3/lib/bashrc.stub >>
.bashrc
```

```
bash-2.05b$ ls -la
total 20
drwxr-xr-x  2 pcs  users  4096 Oct 28 17:43 .
drwxr-xr-x  6 root root   4096 Oct 28 17:18 ..
-rw-----  1 pcs  users   81 Oct 28 17:25
.bash_history
-rw-r--r--  1 pcs  users   905 Oct 28 17:43 .bashrc
-rw-----  1 pcs  users   99 Oct 28 17:23 .rhosts
```

```
bash-2.05b$ pico .bashrc
```

b.3) Agregar al .bashrc lo siguiente:

```
PVM_ROOT=/usr/share/pvm3
PVM_DPATH=$PVM_ROOT/lib/pvmd
export PVM_ROOT PVM_DPATH
```

b.4) descomentar las tres líneas (export) finales

```
if [ -n $PVM_ROOT ]; then
    export PVM_ARCH=`$PVM_ROOT/lib/pvmgetarch`
    #
    # uncomment one of the following lines if you want the
PVM commands
    # directory to be added to your shell path.
    #
        export PATH=$PATH:$PVM_ROOT/lib                #
generic        export PATH=$PATH:$PVM_ROOT/lib/$PVM_ARCH #
arch-specific
    #
    # uncomment the following line if you want the PVM
executable directory
    # to be added to your shell path.
    #
        export PATH=$PATH:$PVM_ROOT/bin/$PVM_ARCH
fi
```

b.5) salir del editor y grabar :)

b.6) ejecutar el .bashrc:

```
bash-2.05b$ . .bashrc
```

4 Ejecutar PVM

```
$pvm
pvm> help
pvm> conf
pvm> add <hostname>
pvm> conf
pvm> quit
```

Si todo está OK el segundo conf debería mostrar el nuevo host agregado

SUGERENCIA IMPORTANTE: Hay mucha más información en el libro “pvm_book.ps” que está en “material en línea” a partir de la página 22.

5 Ejecutar ejemplos con PVM

(ver también la sección 3.6 del libro pvm_book.ps)

a) Copiar el directorio de ejemplos al directorio local (home)

```

$ cd
$ mkdir pvm3
$ cp -r $PVM_ROOT/examples $HOME/pvm3/examples
$ cd pvm3/examples

```

b) Modificar el archivo Makefile.aimk cambiando la referencia al directorio de PVM

```

#PVMDIR      =    ../..
PVMDIR      =    /usr/share/pvm3
SDIR        =    /home/pes/pvm3/examples

```

c) Compilar algunos programas (en todos los hosts que esten en el sistema), por ejemplo:

```

$ aimk hello
$ aimk hello_other

```

d) Ejecutar el ejemplo

```

$pvm
pvm> spawn -> hello

```

si todo está OK el programa hello se ejecutará en un host, y hello_other en otro.

6 Ejercitación sugerida

- Modifique el programa Hello para que ejecute al hijo en un nodo determinado por usted
- Modifique otro programa de ejemplo
- Haga un programa nuevo