



LibreOffice®

PROYECTO 1

- Virtualización, Kernel y Módulos
- Cristian Alonso Ramírez Guzmán
- 200924468 – Sistemas operativos 1

HIPERVISOR KVM

■ Kvm:

■ Kernel-based Virtual Machine o KVM, (en español, Máquina virtual basada en el núcleo) es una solución para implementar virtualización completa con Linux. (fuente: wikipedia)

■ Configuración:

- - Sistema anfitrión: Ubuntu 16.04
- - Hipervisor: KVM kernel-based Virtual Machine
- - Sistema Huesped: Fedora 27

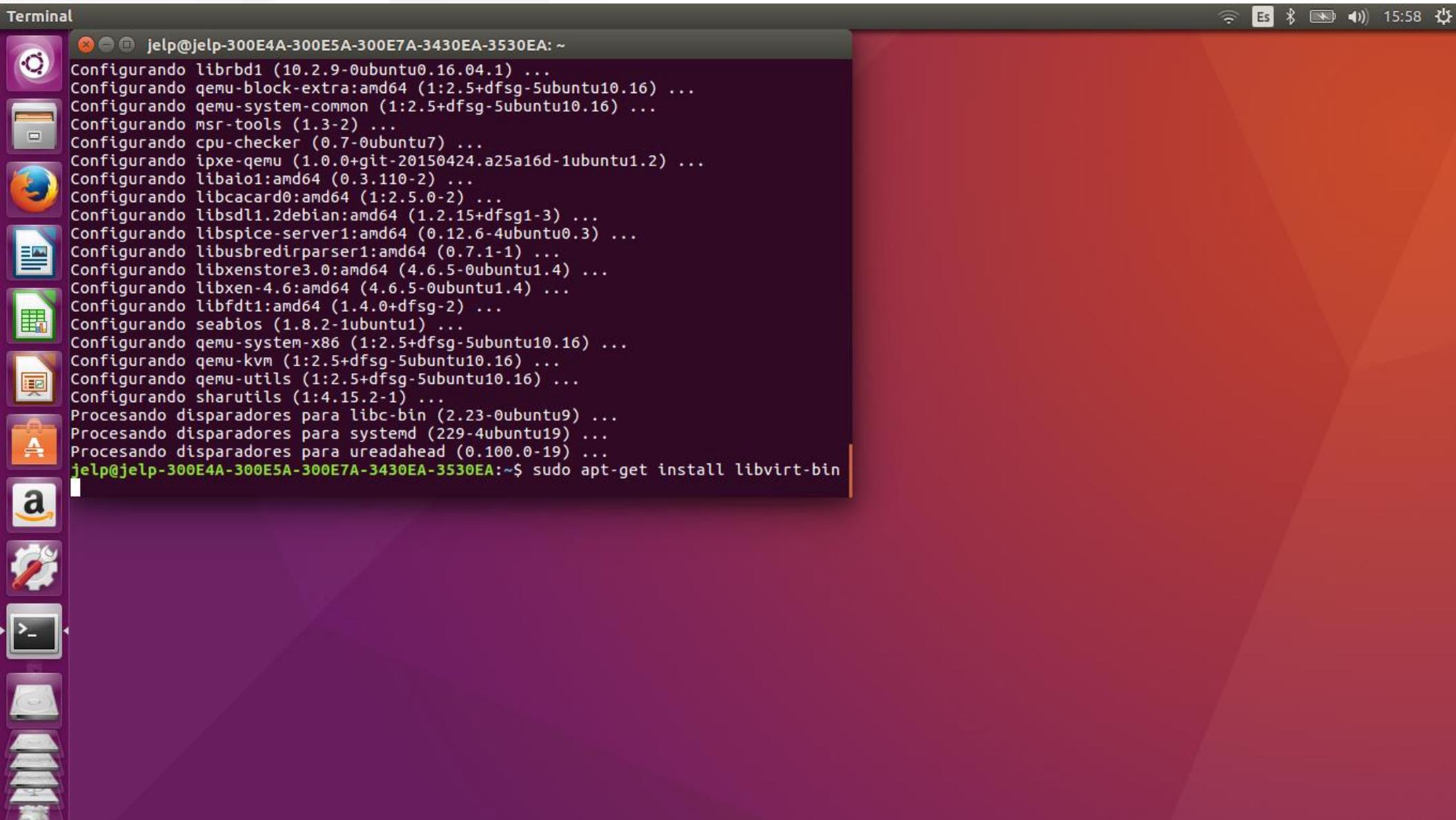


INSTALACION KVM

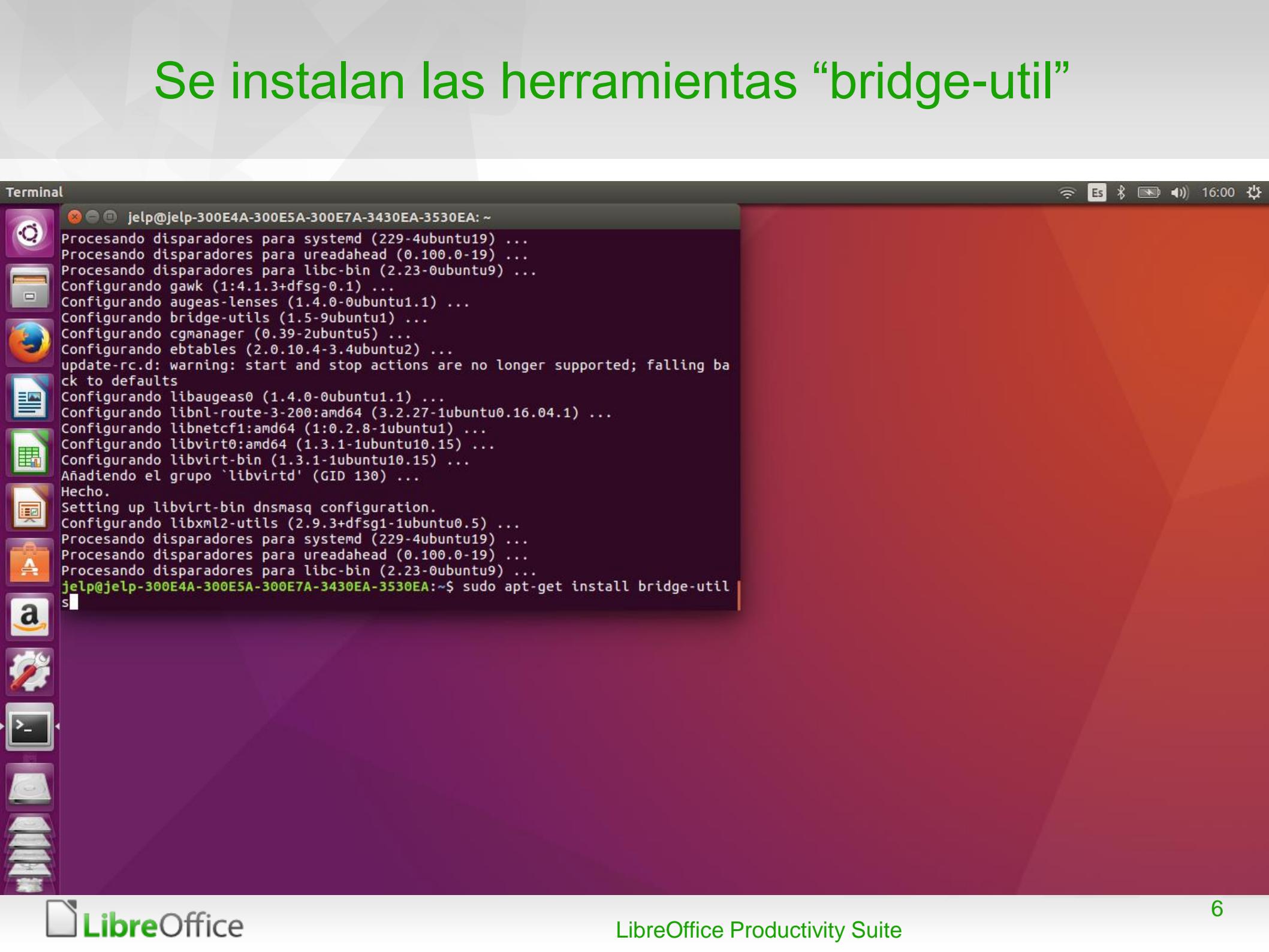
Verificación de virtualización por hardware e instalación de qemu-kvm

```
jelphome@jelphome-OptiPlex-5090:~$ egrep -c 'vmx' /proc/cpuinfo  
4  
jelphome@jelphome-OptiPlex-5090:~$ sudo apt-get install qemu-kvm
```

Se instala el paquete binario libvirt, que permite virtualizar.



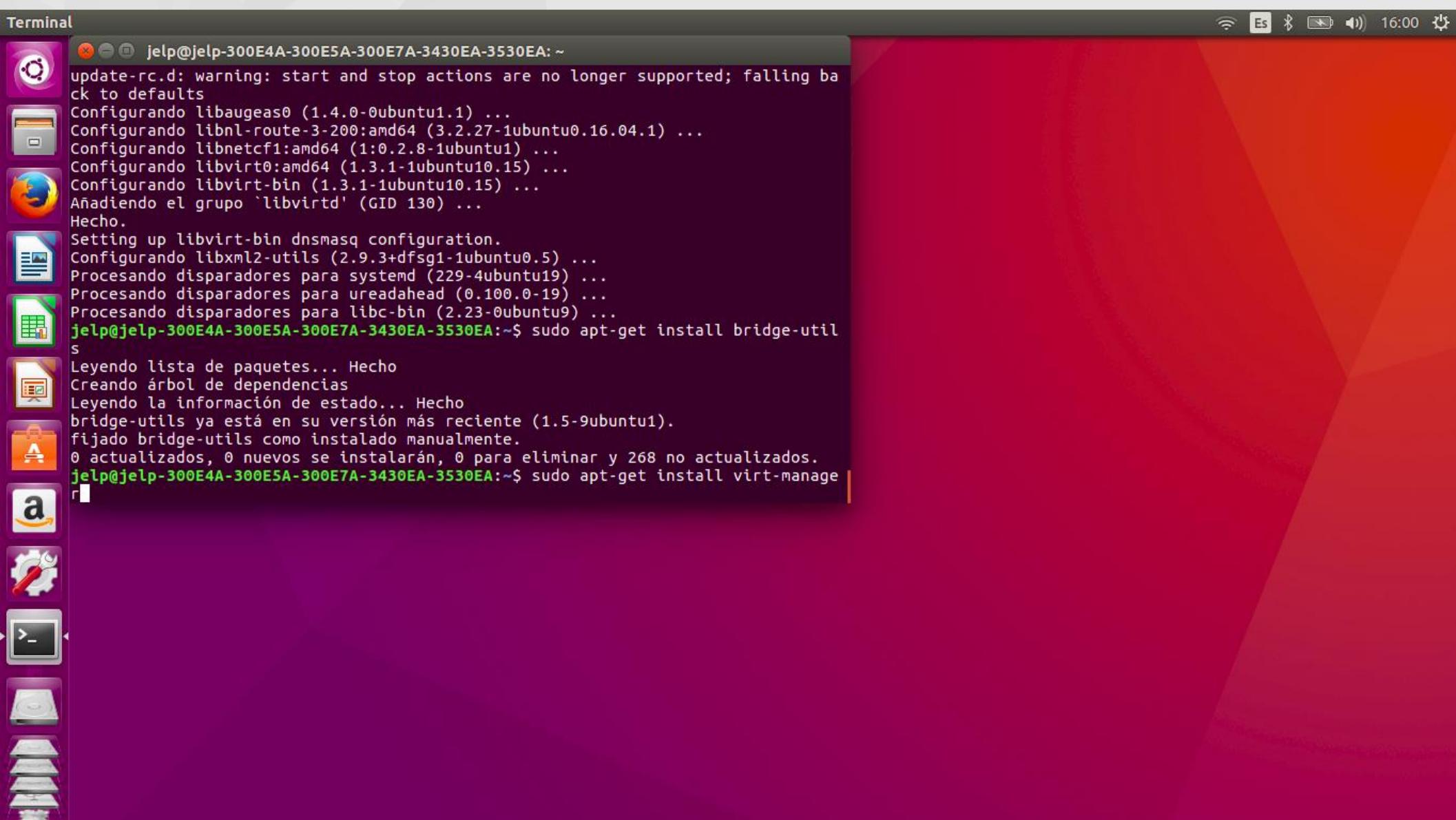
Se instalan las herramientas “bridge-util”



A screenshot of an Ubuntu desktop environment. A terminal window is open in the top-left corner, showing the output of a command to install bridge-util. The terminal window title is "Terminal" and the user is "jelp@jelp-300E4A-300E5A-300E7A-3430EA-3530EA". The terminal shows the following text:

```
jelp@jelp-300E4A-300E5A-300E7A-3430EA-3530EA: ~
Procesando disparadores para systemd (229-4ubuntu19) ...
Procesando disparadores para ureadahead (0.100.0-19) ...
Procesando disparadores para libc-bin (2.23-0ubuntu9) ...
Configurando gawk (1:4.1.3+dfsg-0.1) ...
Configurando augeas-lenses (1.4.0-0ubuntu1.1) ...
Configurando bridge-utils (1.5-9ubuntu1) ...
Configurando cgmanager (0.39-2ubuntu5) ...
Configurando ebttables (2.0.10.4-3.4ubuntu2) ...
update-rc.d: warning: start and stop actions are no longer supported; falling back to defaults
Configurando libaugeas0 (1.4.0-0ubuntu1.1) ...
Configurando libnl-route-3-200:amd64 (3.2.27-1ubuntu0.16.04.1) ...
Configurando libnetcf1:amd64 (1:0.2.8-1ubuntu1) ...
Configurando libvirt0:amd64 (1.3.1-1ubuntu10.15) ...
Configurando libvirt-bin (1.3.1-1ubuntu10.15) ...
Añadiendo el grupo `libvirtd' (GID 130) ...
Hecho.
Setting up libvirt-bin dnsmasq configuration.
Configurando libxml2-utils (2.9.3+dfsg1-1ubuntu0.5) ...
Procesando disparadores para systemd (229-4ubuntu19) ...
Procesando disparadores para ureadahead (0.100.0-19) ...
Procesando disparadores para libc-bin (2.23-0ubuntu9) ...
jelp@jelp-300E4A-300E5A-300E7A-3430EA-3530EA:~$ sudo apt-get install bridge-util
```

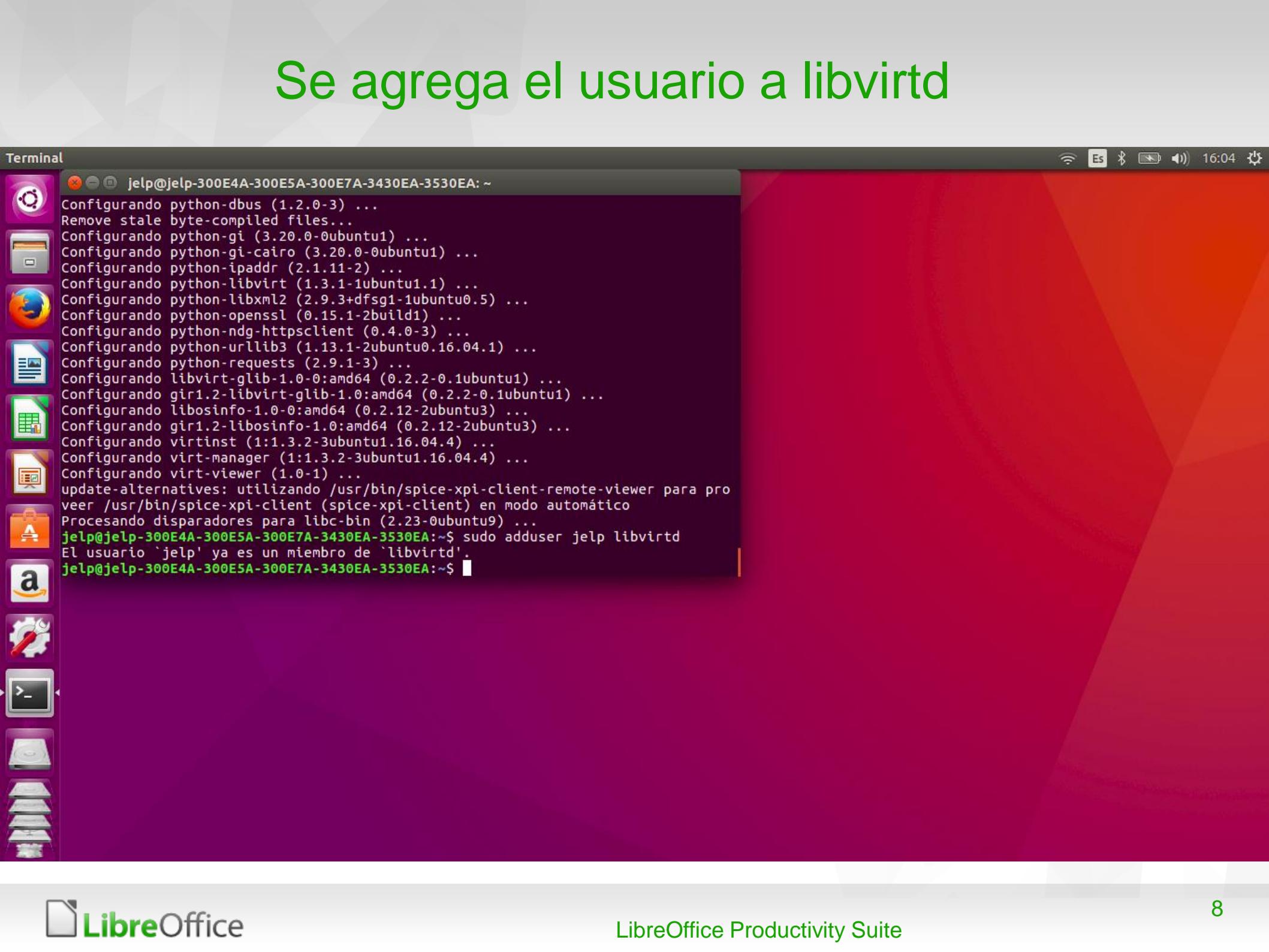
Se instala el manager del virtualizador “virt-manager”.



A screenshot of an Ubuntu desktop environment. A terminal window is open in the top-left corner, showing the command line interface. The terminal window title is "Terminal". The command being run is "sudo apt-get install virt-manager". The output of the command shows various package configurations and dependencies being processed. The desktop background is red, and the Unity interface is visible with its characteristic icons and dock.

```
jelp@jelp-300E4A-300E5A-300E7A-3430EA-3530EA: ~
update-rc.d: warning: start and stop actions are no longer supported; falling back to defaults
Configurando libaugeas0 (1.4.0-0ubuntu1.1) ...
Configurando libnl-route-3-200:amd64 (3.2.27-1ubuntu0.16.04.1) ...
Configurando libnetcf1:amd64 (1:0.2.8-1ubuntu1) ...
Configurando libvirt0:amd64 (1.3.1-1ubuntu10.15) ...
Configurando libvirt-bin (1.3.1-1ubuntu10.15) ...
Añadiendo el grupo `libvirtd' (GID 130) ...
Hecho.
Setting up libvirt-bin dnsmasq configuration.
Configurando libxml2-utils (2.9.3+dfsg1-1ubuntu0.5) ...
Procesando disparadores para systemd (229-4ubuntu19) ...
Procesando disparadores para ureadahead (0.100.0-19) ...
Procesando disparadores para libc-bin (2.23-0ubuntu9) ...
jelp@jelp-300E4A-300E5A-300E7A-3430EA-3530EA:~$ sudo apt-get install bridge-utils
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
bridge-utils ya está en su versión más reciente (1.5-9ubuntu1).
fijado bridge-utils como instalado manualmente.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 268 no actualizados.
jelp@jelp-300E4A-300E5A-300E7A-3430EA-3530EA:~$ sudo apt-get install virt-manag
r
```

Se agrega el usuario a libvirtd

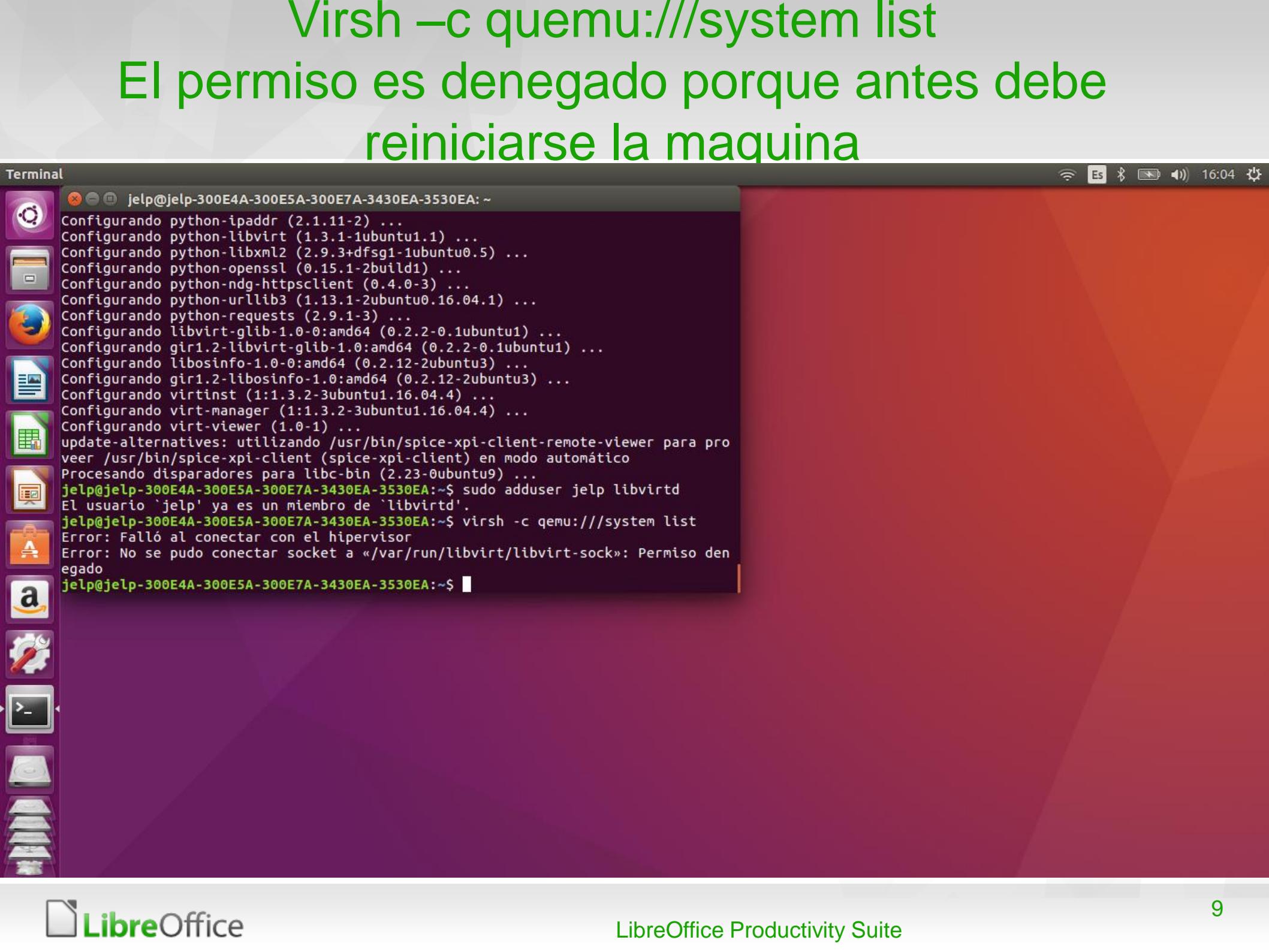


A screenshot of an Ubuntu desktop environment. A terminal window titled "Terminal" is open at the top left, showing the command line. The desktop background is red and orange. The system tray at the top right shows icons for battery, signal, and volume.

```
jelp@jelp-300E4A-300E5A-300E7A-3430EA-3530EA: ~
Configurando python-dbus (1.2.0-3) ...
Remove stale byte-compiled files...
Configurando python-gi (3.20.0-0ubuntu1) ...
Configurando python-gi-cairo (3.20.0-0ubuntu1) ...
Configurando python-ipaddr (2.1.11-2) ...
Configurando python-libvirt (1.3.1-1ubuntu1.1) ...
Configurando python-libxml2 (2.9.3+dfsg1-1ubuntu0.5) ...
Configurando python-openssl (0.15.1-2build1) ...
Configurando python-ndg-httpsclient (0.4.0-3) ...
Configurando python-urllib3 (1.13.1-2ubuntu0.16.04.1) ...
Configurando python-requests (2.9.1-3) ...
Configurando libvirt-glib-1.0-0:amd64 (0.2.2-0.1ubuntu1) ...
Configurando gir1.2-libvirt-glib-1.0:amd64 (0.2.2-0.1ubuntu1) ...
Configurando libosinfo-1.0-0:amd64 (0.2.12-2ubuntu3) ...
Configurando gir1.2-libosinfo-1.0:amd64 (0.2.12-2ubuntu3) ...
Configurando virtinst (1:1.3.2-3ubuntu1.16.04.4) ...
Configurando virt-manager (1:1.3.2-3ubuntu1.16.04.4) ...
Configurando virt-viewer (1.0-1) ...
update-alternatives: utilizando /usr/bin/spice-xpi-client-remote-viewer para pro
veer /usr/bin/spice-xpi-client (spice-xpi-client) en modo automático
Procesando disparadores para libc-bin (2.23-0ubuntu9) ...
jelp@jelp-300E4A-300E5A-300E7A-3430EA-3530EA:~$ sudo adduser jelp libvirtd
El usuario 'jelp' ya es un miembro de 'libvirtd'.
jelp@jelp-300E4A-300E5A-300E7A-3430EA-3530EA:~$
```

Virsh –c qemu:///system list

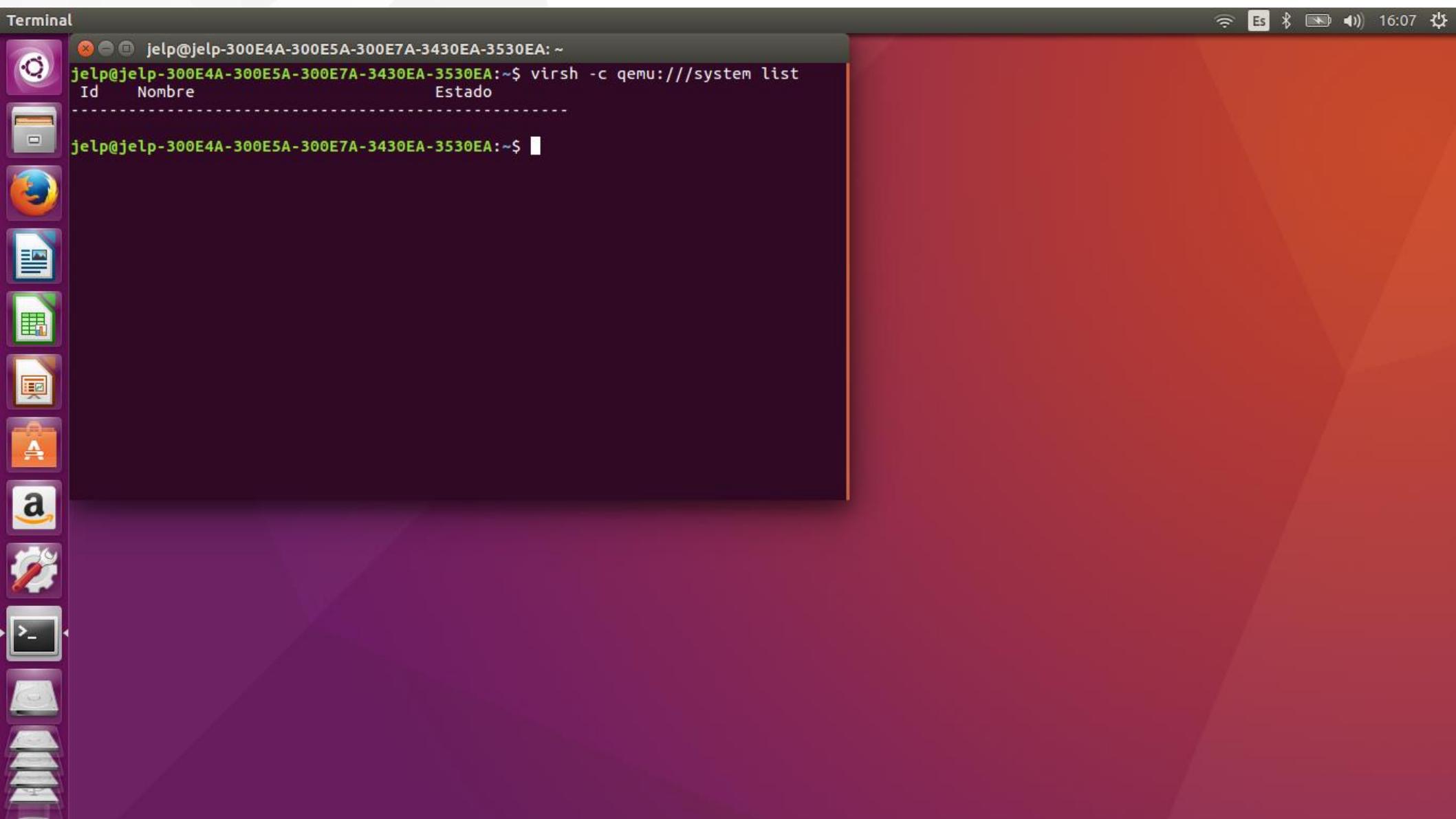
El permiso es denegado porque antes debe reiniciarse la maquina



A screenshot of an Ubuntu desktop environment. A terminal window is open in the top-left corner, showing the following command-line session:

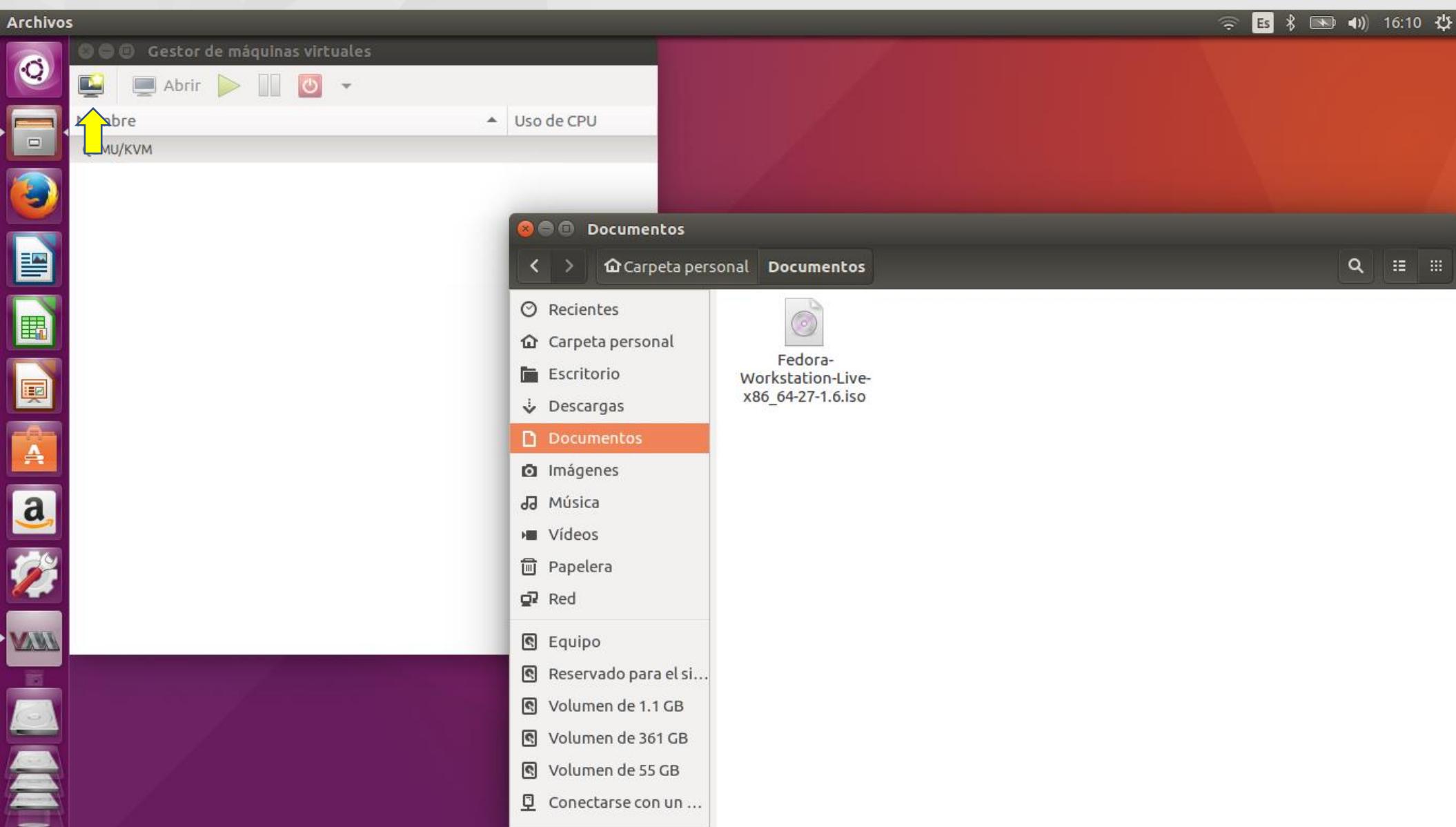
```
jelp@jelp-300E4A-300E5A-300E7A-3430EA-3530EA: ~
Configurando python-ipaddr (2.1.11-2) ...
Configurando python-libvirt (1.3.1-1ubuntu1.1) ...
Configurando python-libxml2 (2.9.3+dfsg1-1ubuntu0.5) ...
Configurando python-openssl (0.15.1-2build1) ...
Configurando python-ndg-httpsclient (0.4.0-3) ...
Configurando python-urllib3 (1.13.1-2ubuntu0.16.04.1) ...
Configurando python-requests (2.9.1-3) ...
Configurando libvirt-glib-1.0-0:amd64 (0.2.2-0.1ubuntu1) ...
Configurando gir1.2-libvirt-glib-1.0:amd64 (0.2.2-0.1ubuntu1) ...
Configurando libosinfo-1.0-0:amd64 (0.2.12-2ubuntu3) ...
Configurando gir1.2-libosinfo-1.0:amd64 (0.2.12-2ubuntu3) ...
Configurando virtinst (1:1.3.2-3ubuntu1.16.04.4) ...
Configurando virt-manager (1:1.3.2-3ubuntu1.16.04.4) ...
Configurando virt-viewer (1.0-1) ...
update-alternatives: utilizando /usr/bin/spice-xpi-client-remote-viewer para pro
veer /usr/bin/spice-xpi-client (spice-xpi-client) en modo automático
Procesando disparadores para libc-bin (2.23-0ubuntu9) ...
jelp@jelp-300E4A-300E5A-300E7A-3430EA-3530EA:~$ sudo adduser jelp libvirtd
El usuario 'jelp' ya es un miembro de 'libvirtd'.
jelp@jelp-300E4A-300E5A-300E7A-3430EA-3530EA:~$ virsh -c qemu:///system list
Error: Falló al conectar con el hipervisor
Error: No se pudo conectar socket a «/var/run/libvirt/libvirt-sock»: Permiso den
egado
jelp@jelp-300E4A-300E5A-300E7A-3430EA-3530EA:~$
```

Después de reiniciar se ejecuta correctamente el comando.

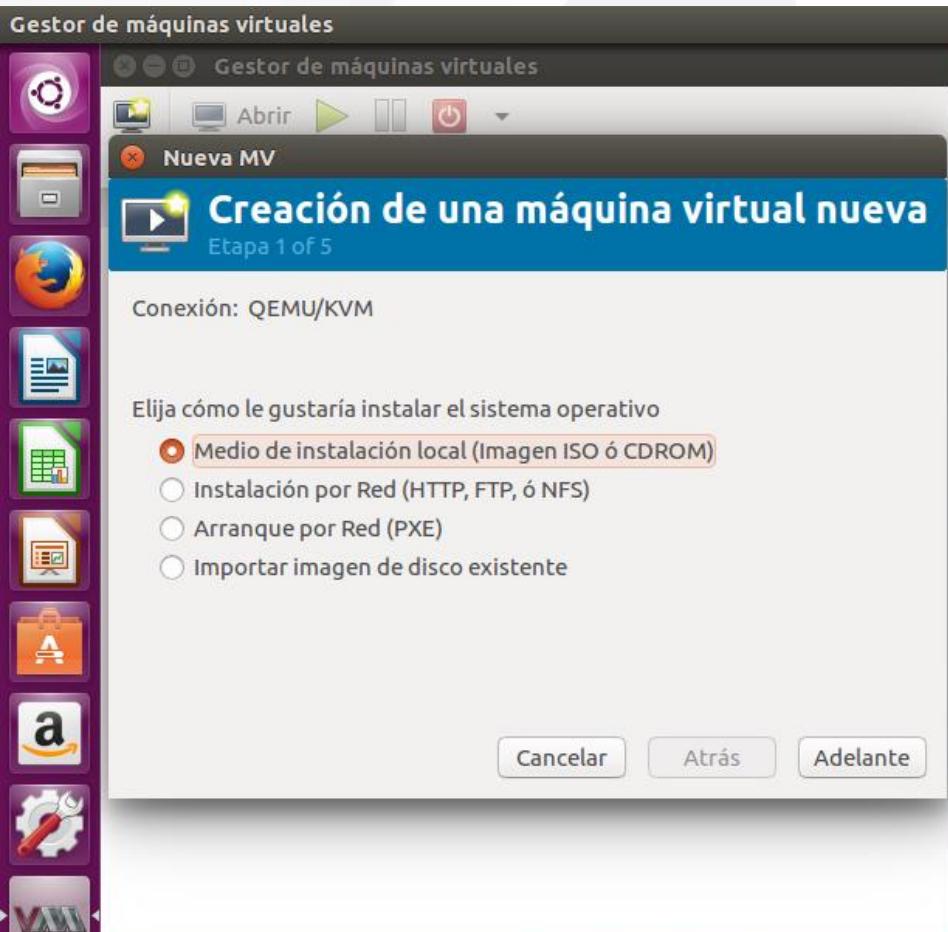


EJECUTANDO KVM

Seleccionamos el botón de “NUEVO”

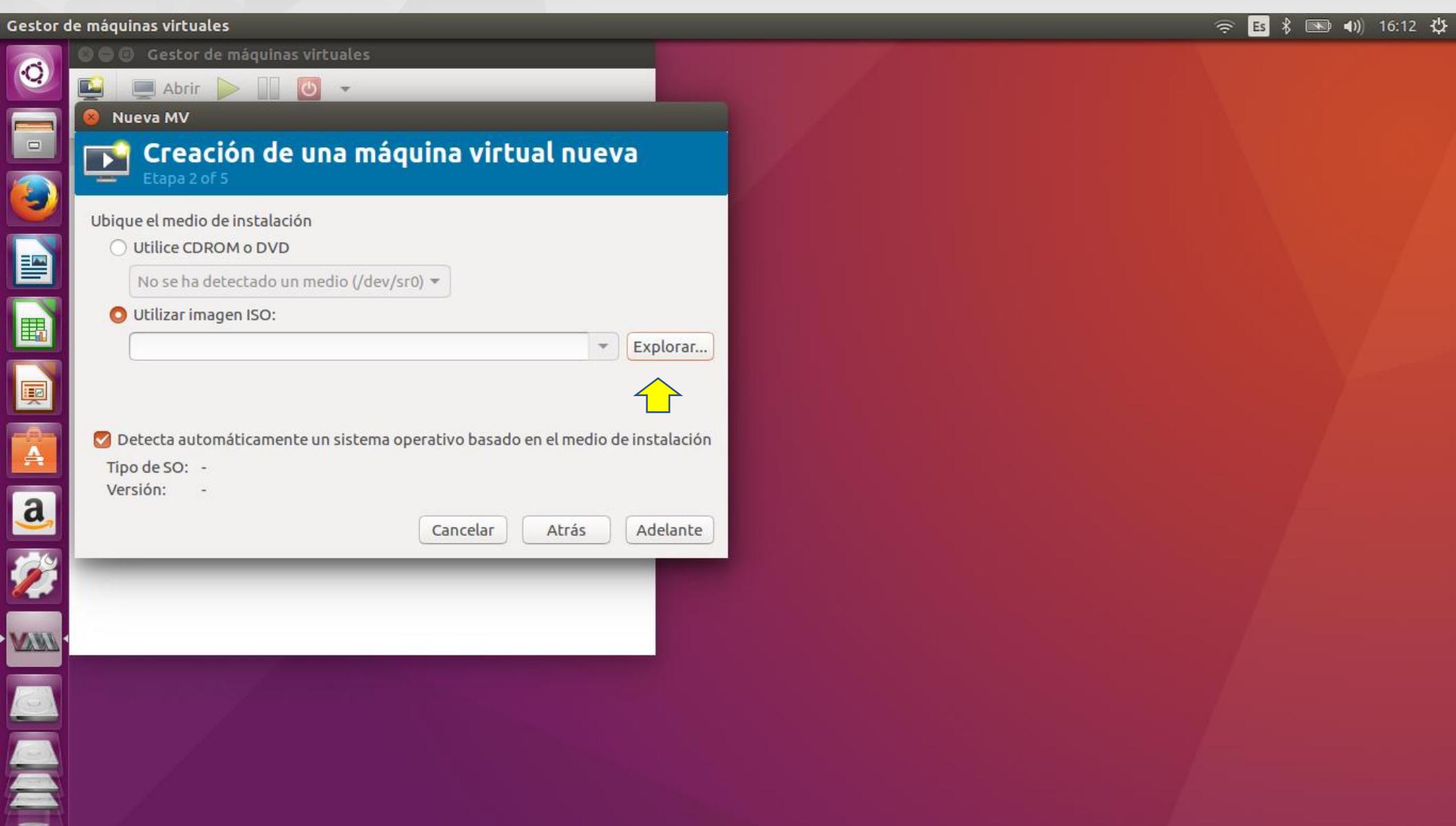


Seleccionar el medio de instalación, en este caso imagen ISO.

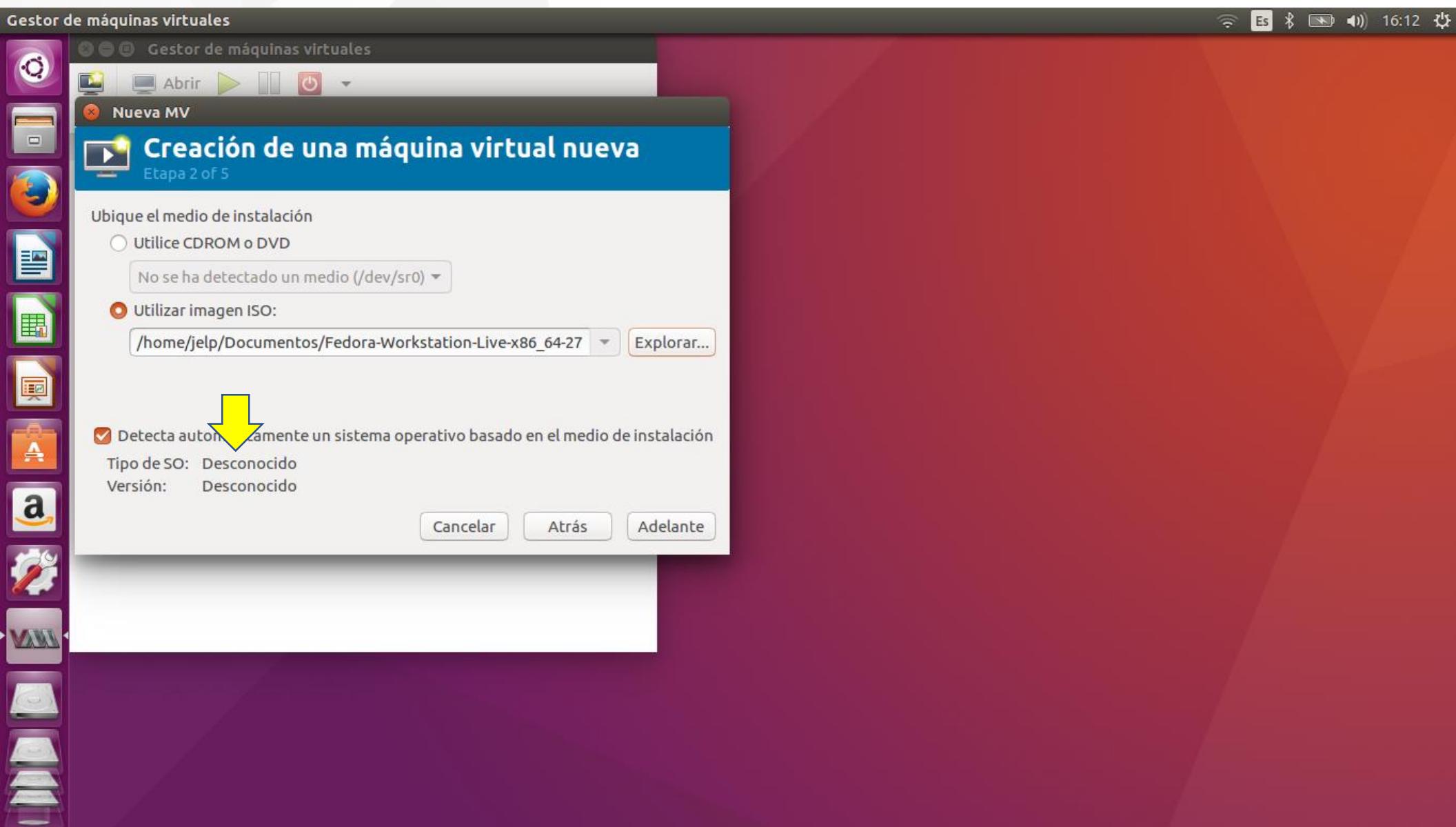


16:11

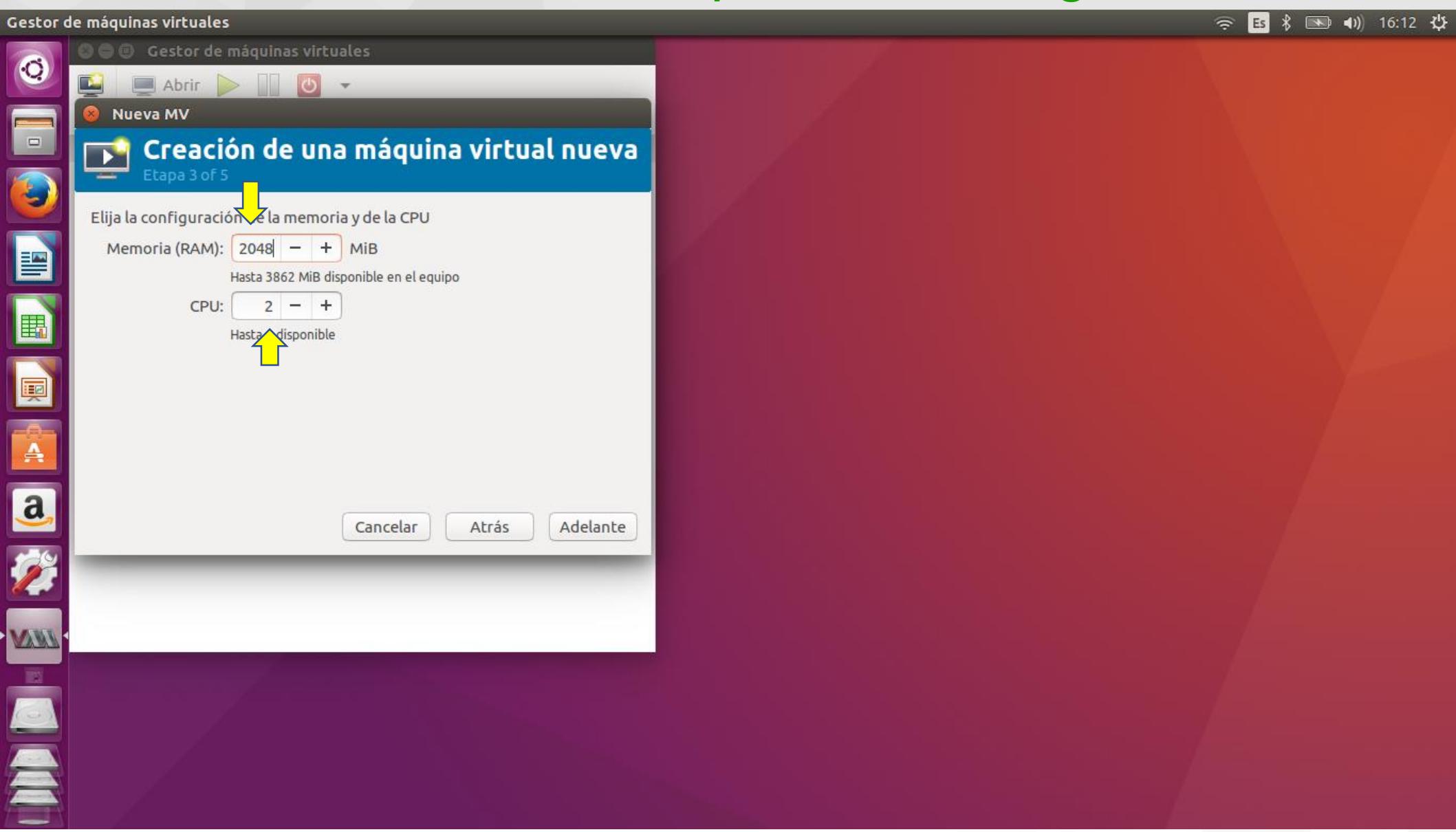
Seleccionamos el botón de “Explorar” para seleccionar la imagen ISO.



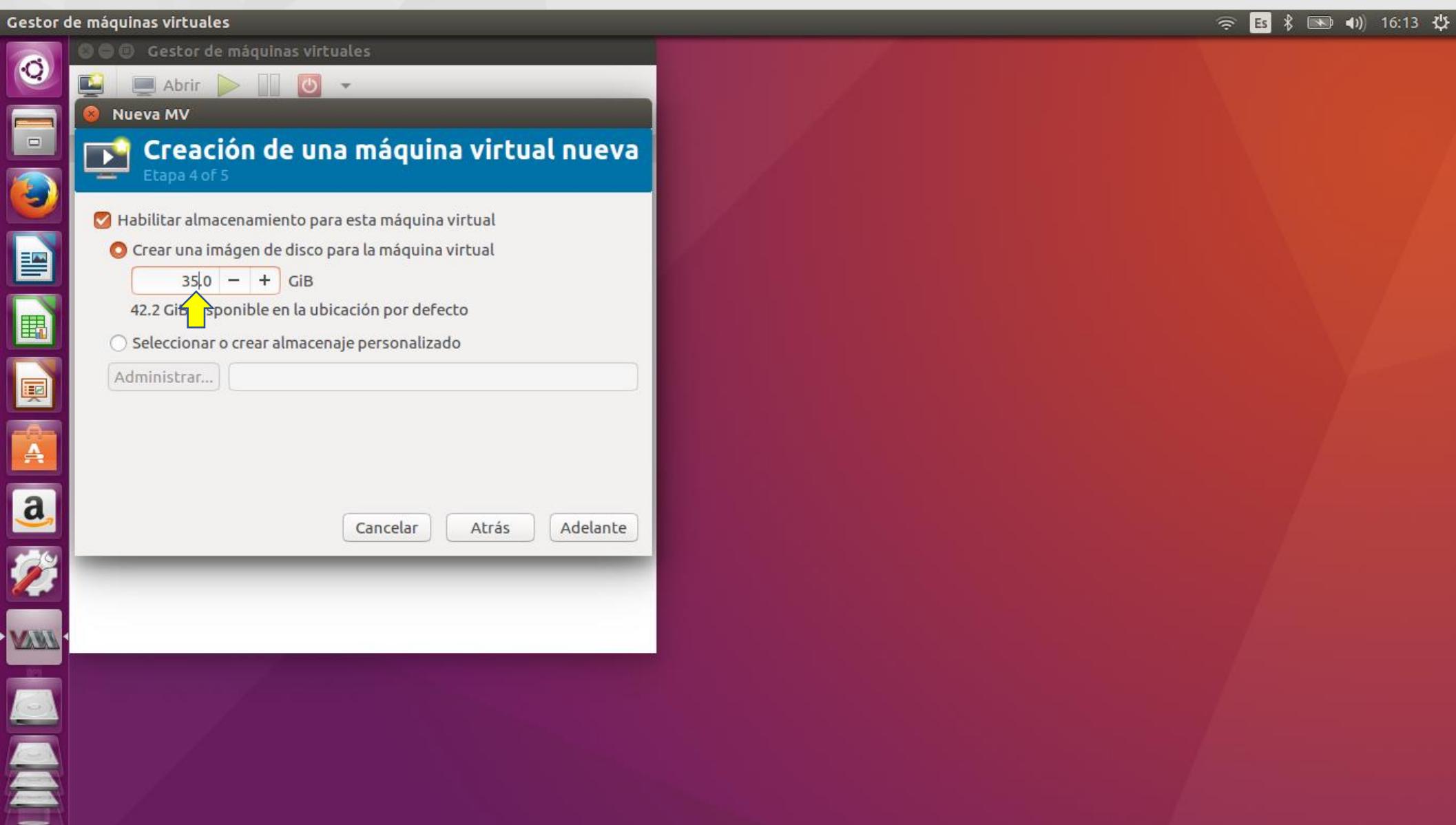
Una vez seleccionada la imagen, KVM tratará de identificar automáticamente el Tipo de Sistema y la versión, si no lo encuentra, usará “Desconocido”.



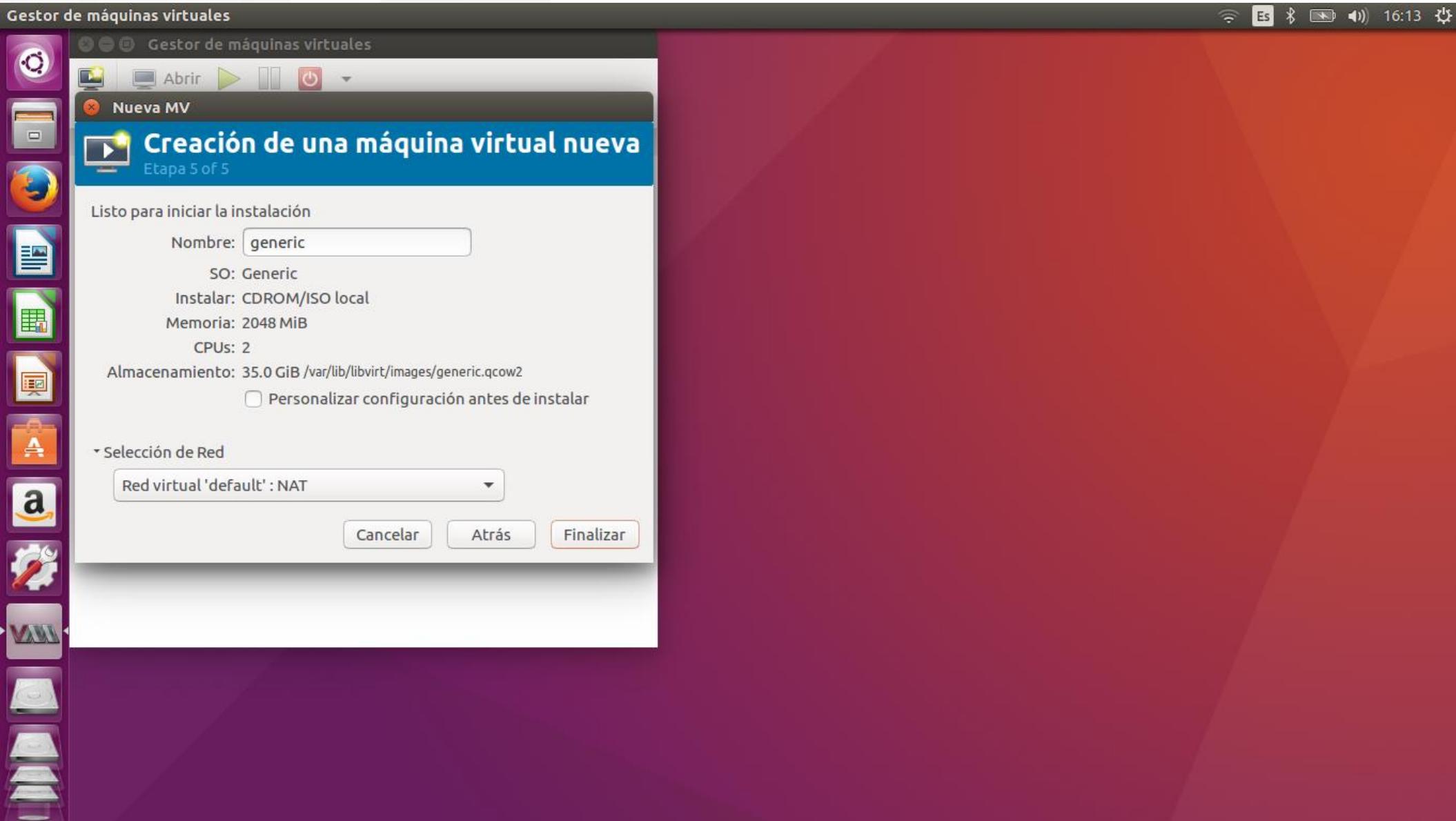
Asignar el tamaño de RAM a utilizar junto con la cantidad de núcleos que desea asignarse.



Asignar el tamaño en Gb. (Recomendado más de 25Gb)



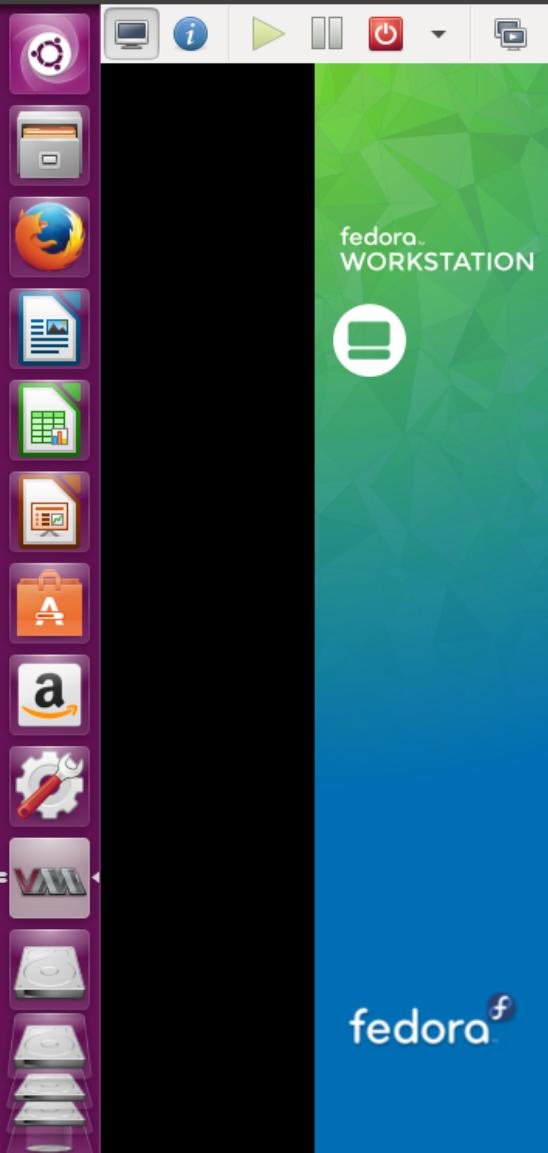
Agregar el nombre de la maquina o nombre de dominio. En este caso “generic”.



INICIANDO INSTALACION DE FEDORA 27 (imagen ISO)

Instalación típica de Sistema operativo





BIENVENIDO A FEDORA 27.

¿Qué idioma quiere utilizar durante el proceso de instalación?

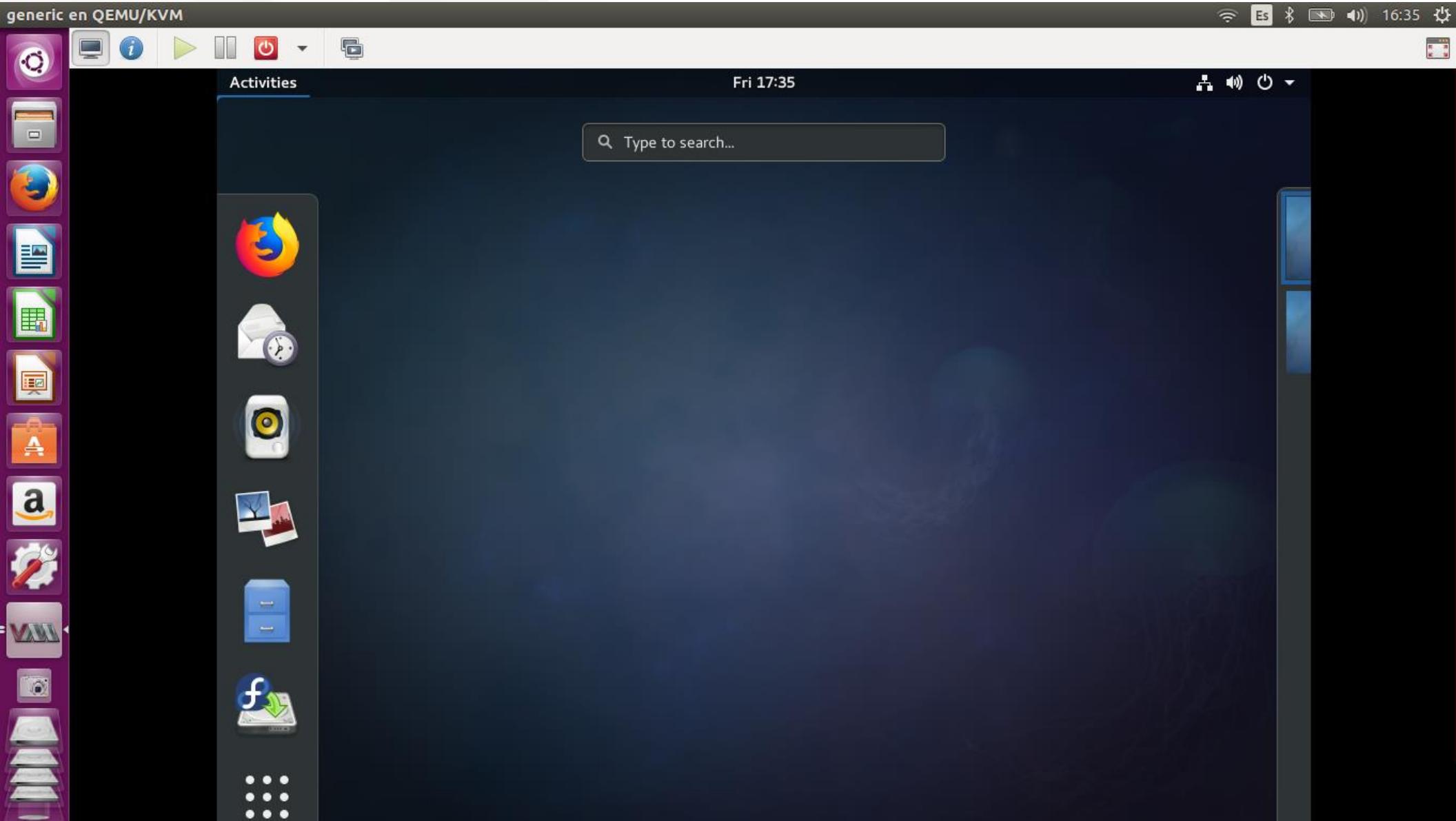
Español	Spanish
Afrikaans	Afrikaans
አማርኛ	Amharic
ଓংগিকা	Angika
العربية	Arabic
অসমীয়া	Assamese
Asturianu	Asturian
Беларуская	Belarusian
Български	Bulgarian
বাংলা	Bangla
ଶ୍ରୀମଦ୍	Tibetan
Brezhoneg	Breton
ବଡ଼ୋ	Bodo
Bosanski	Bosnian

Español (Uruguay)
Español (Puerto Rico)
Español (Perú)
Español (Panamá)
Español (Nicaragua)
Español (México)
Español (Honduras)
Español (Guatemala)
Español (Ecuador)
Español (República Dominicana)
Español (Cuba)
Español (Costa Rica)
Español (Colombia)
Español (Chile)
Español (Bolivia)
Español (Argentina)

Salir

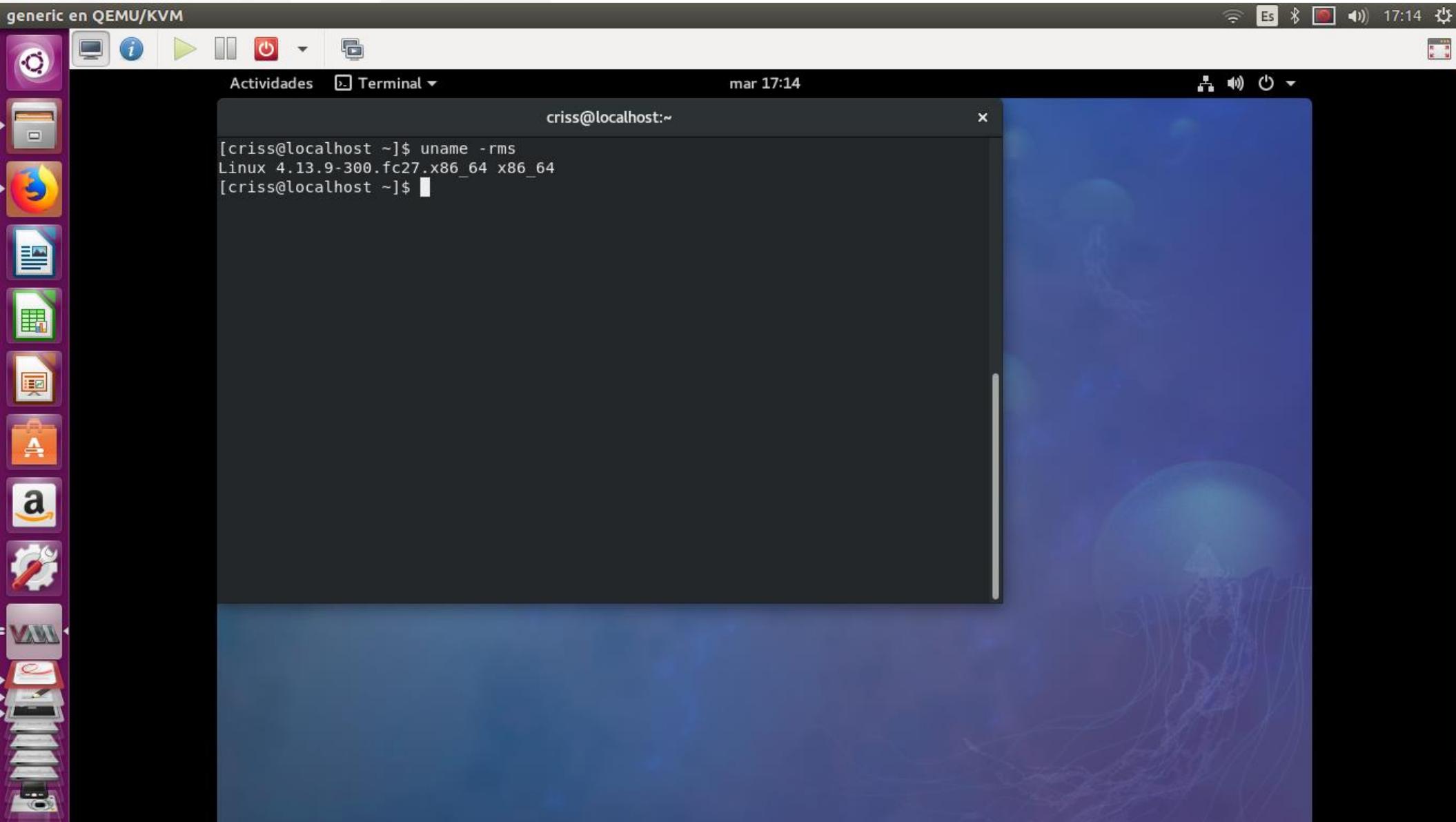
Continuar

Una vez instalado el Sistema operativo, KVM virtualize de la manera que se ve en esta imagen.



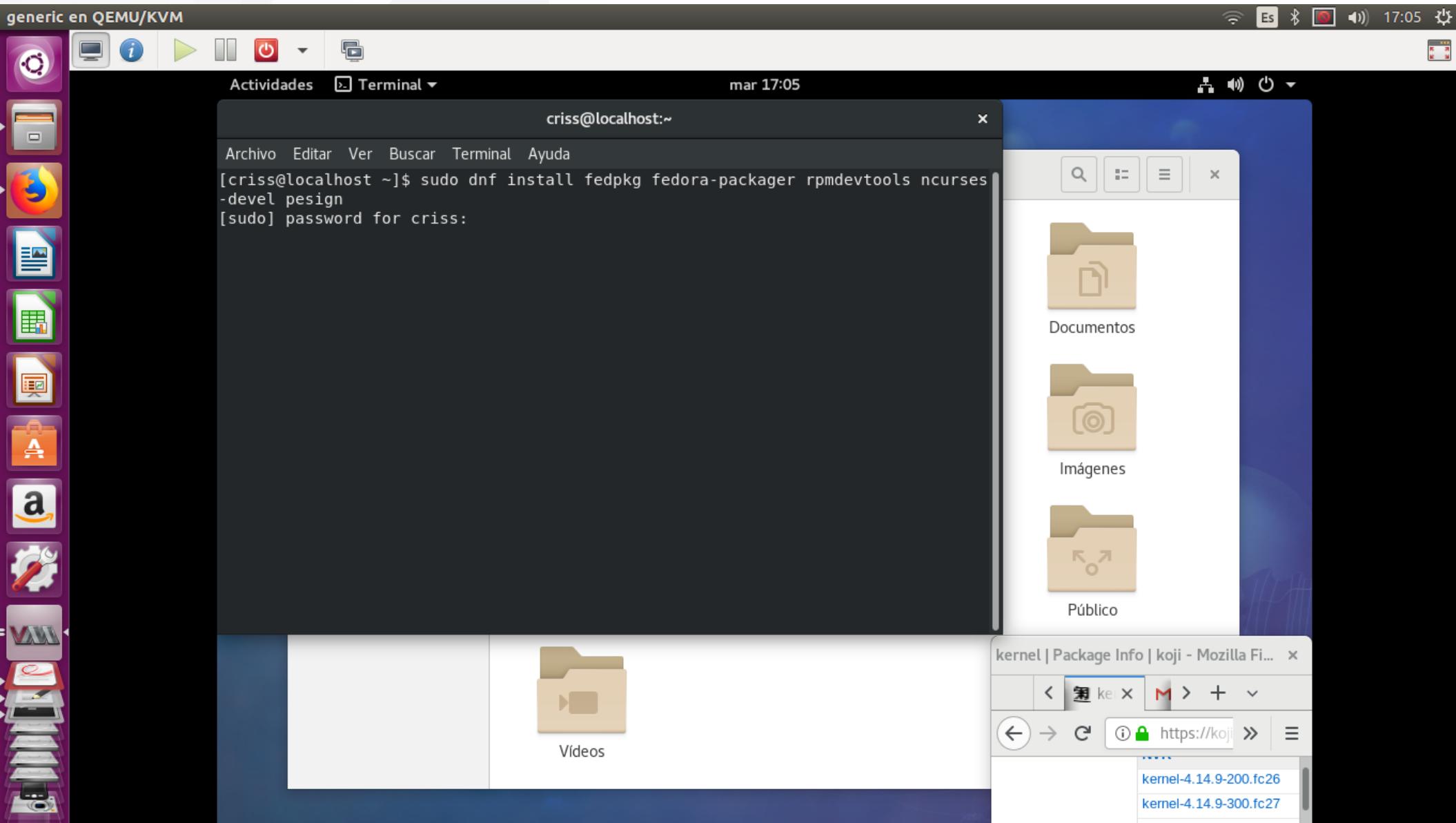
COMPILEACIÓN DE NUEVO KERNEL SOBRE FEDORA 27

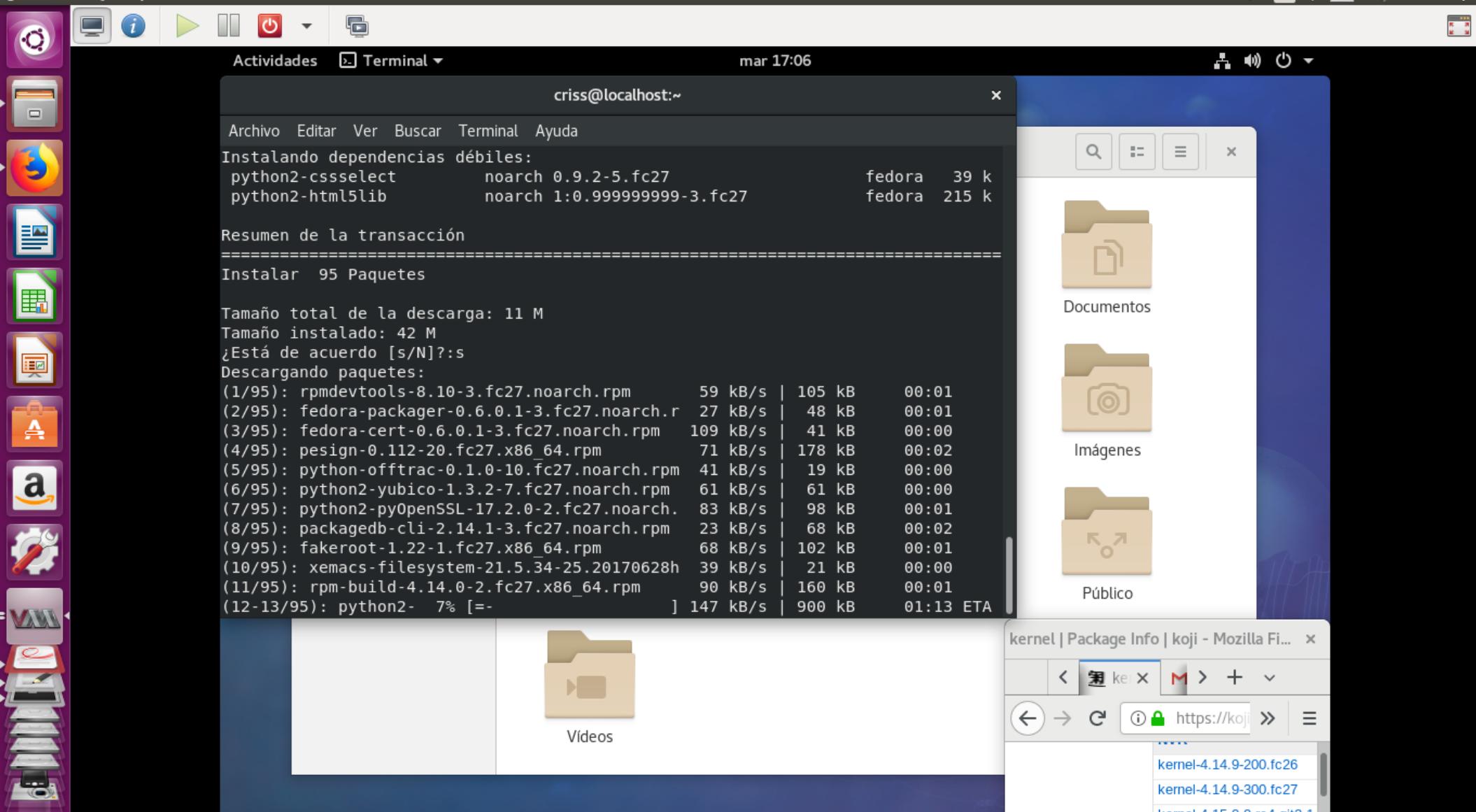
Con el comando “uname –rms” veremos la versión actual del kernel.

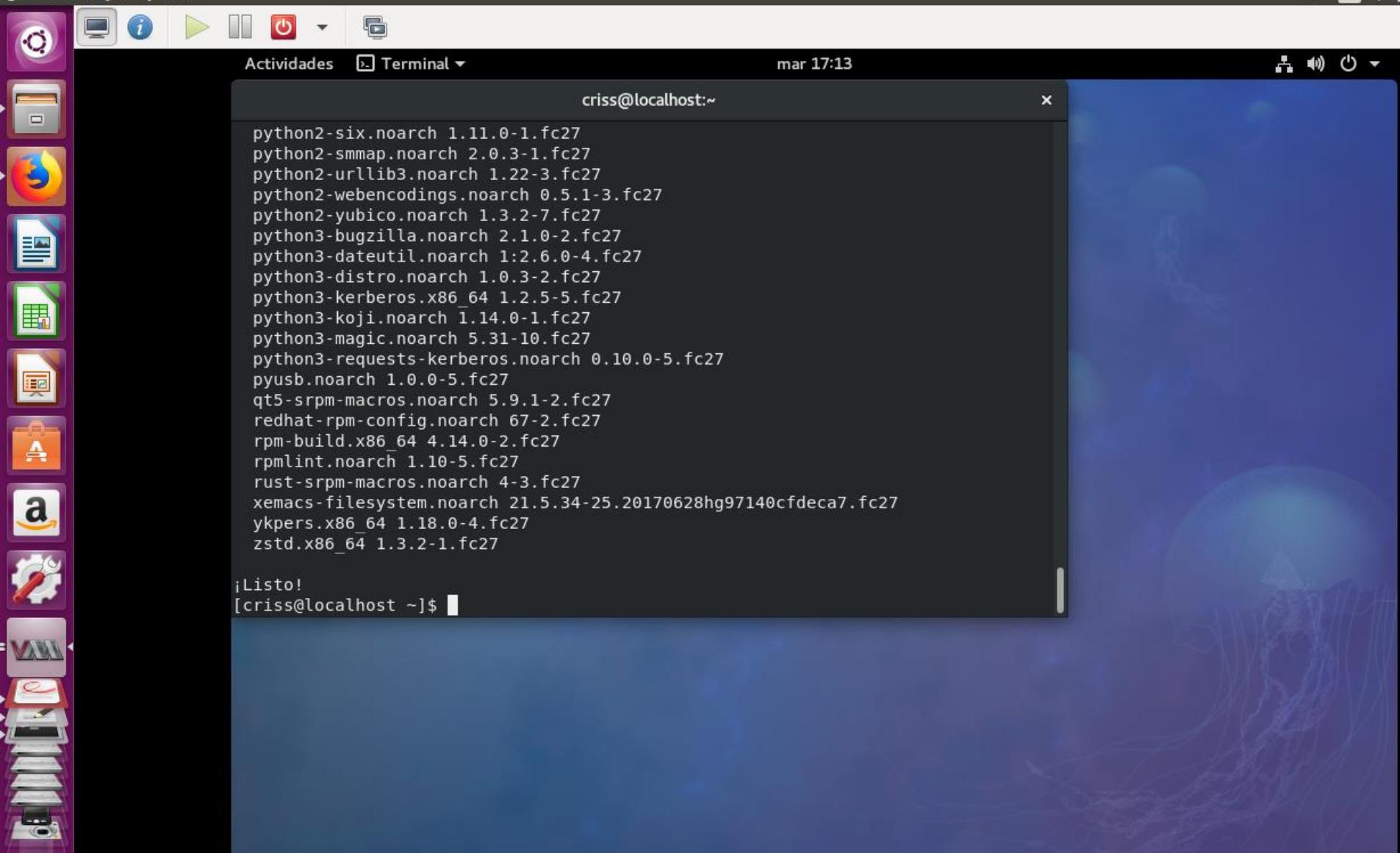


Instalamos las herramientas:

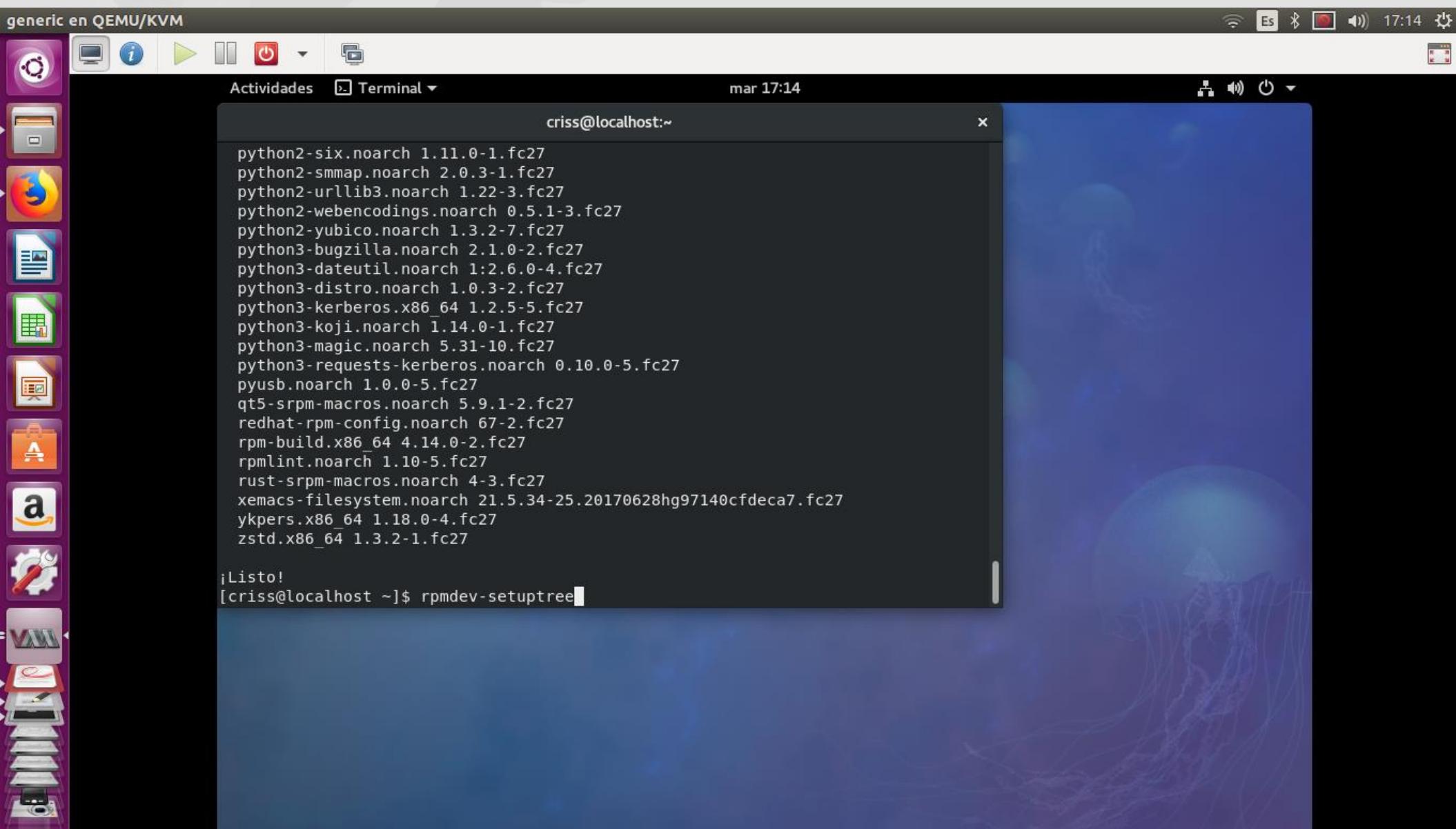
dnf install fedpkg fedora-packager rpmdevtools ncurses-devel pesign





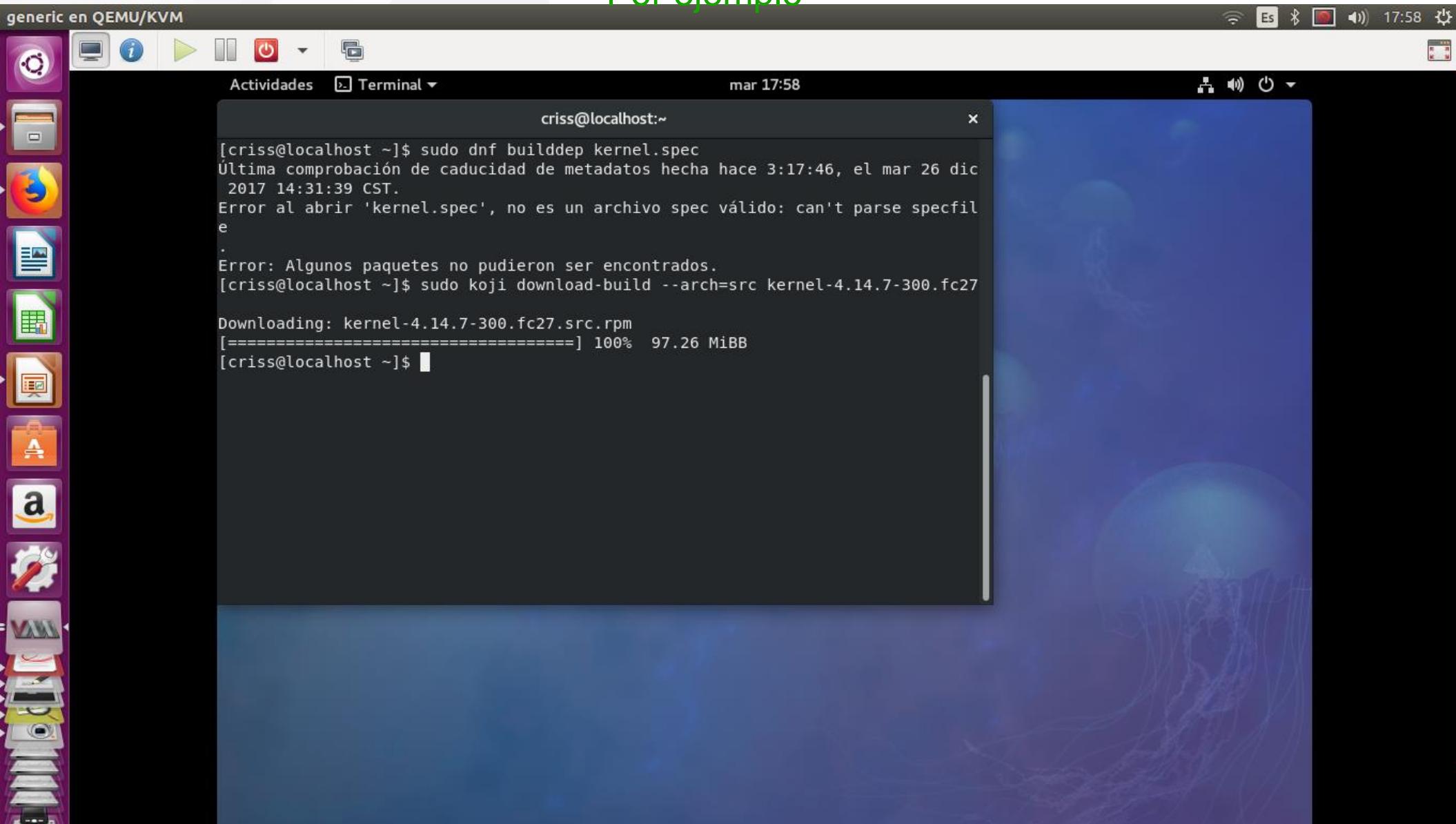


Instalamos “rpmdev-setuptree” para que construya un árbol de carpetas que contendrán archivos que se generarán en el proceso de compilación.



Descargamos las fuentes del nuevo kernel:

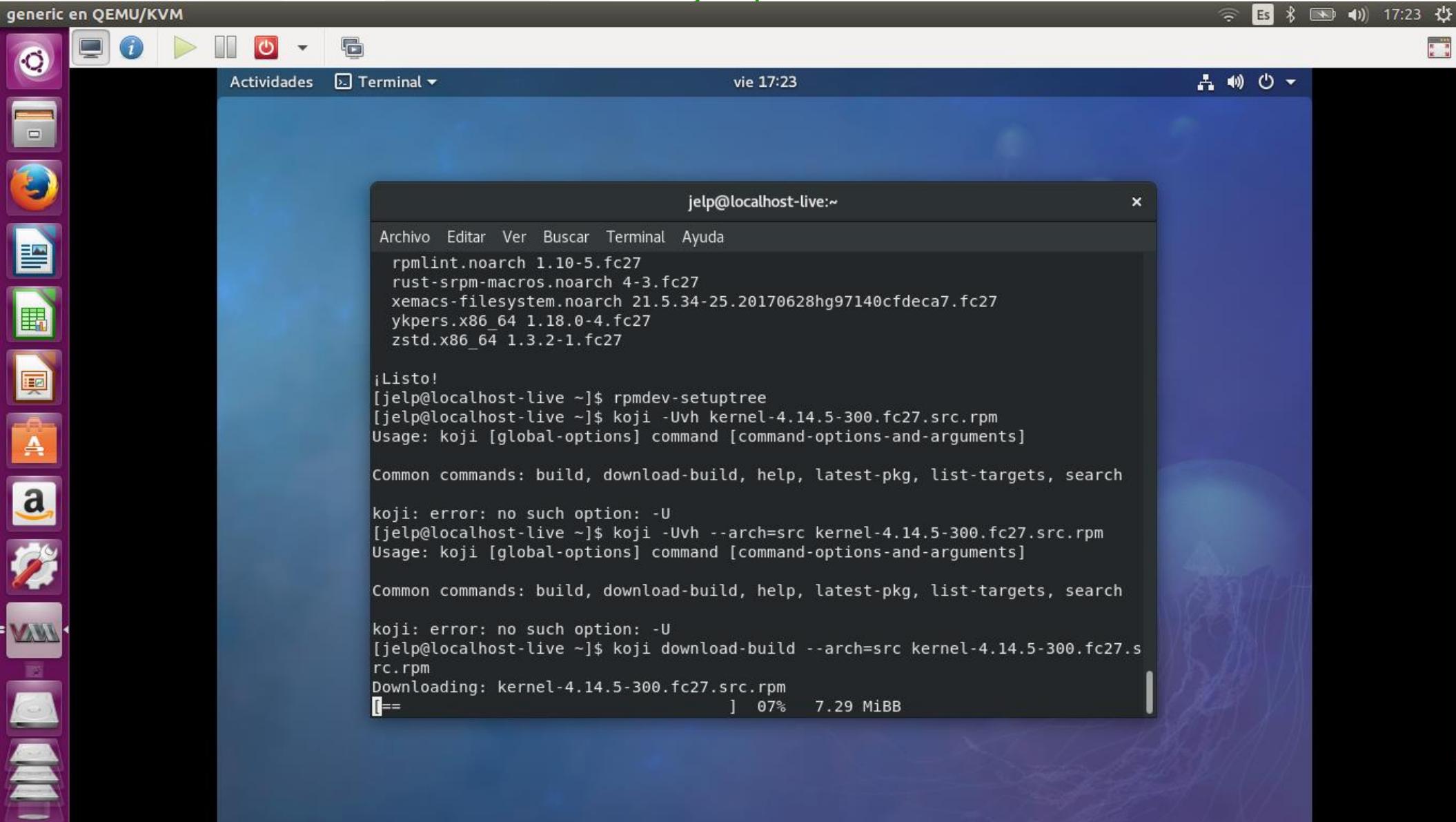
Koji download-build –arch=src kernel-4.14.7.300.fc27 ó Koji download-build –arch=src kernel-4.14.5.300.fc27
Por ejemplo



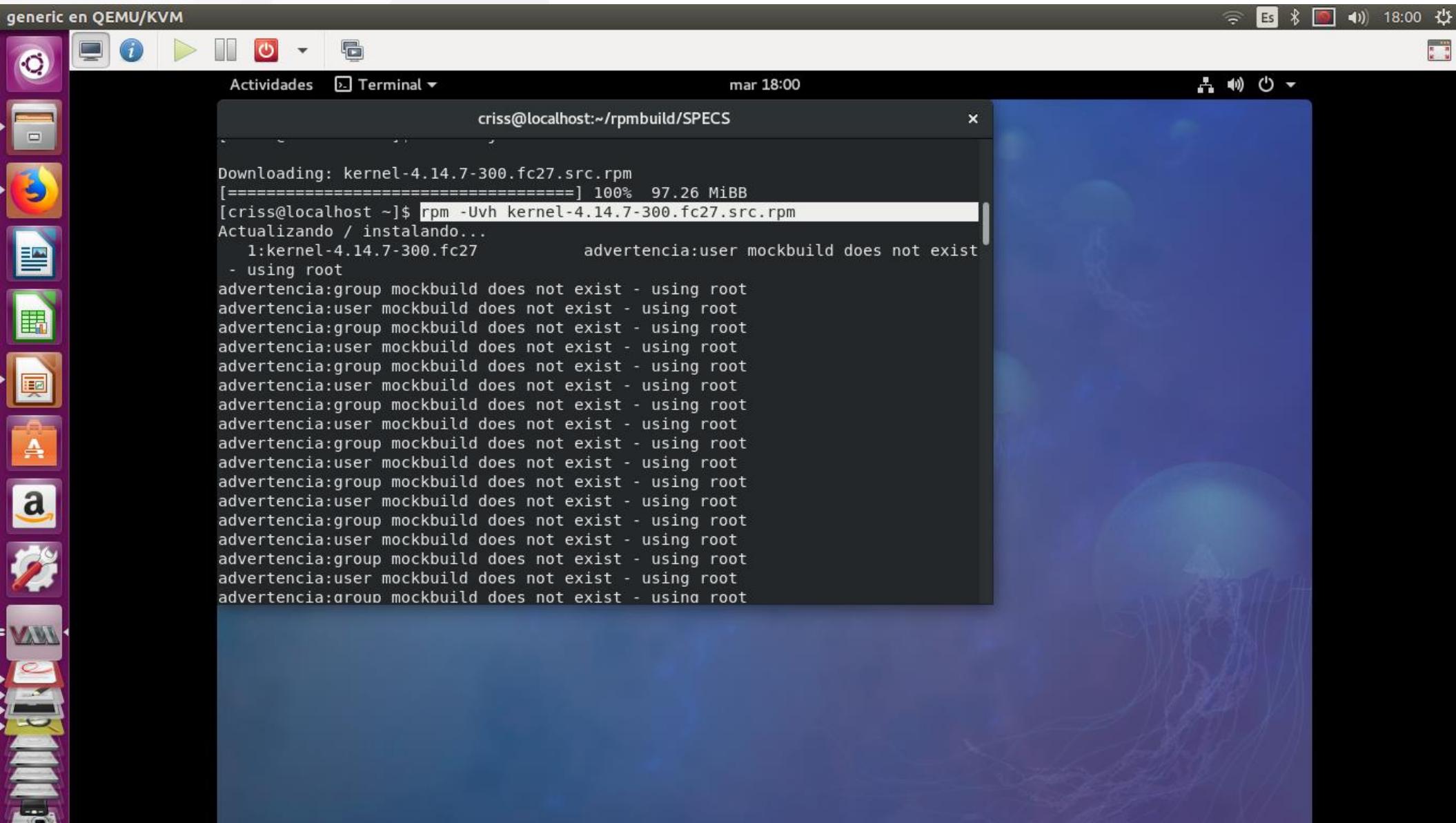
Descargamos las fuentes del nuevo kernel:

Koji download-build –arch=src kernel-4.14.7.300.fc27 ó **Koji download-build –arch=src kernel-4.14.5.300.fc27**

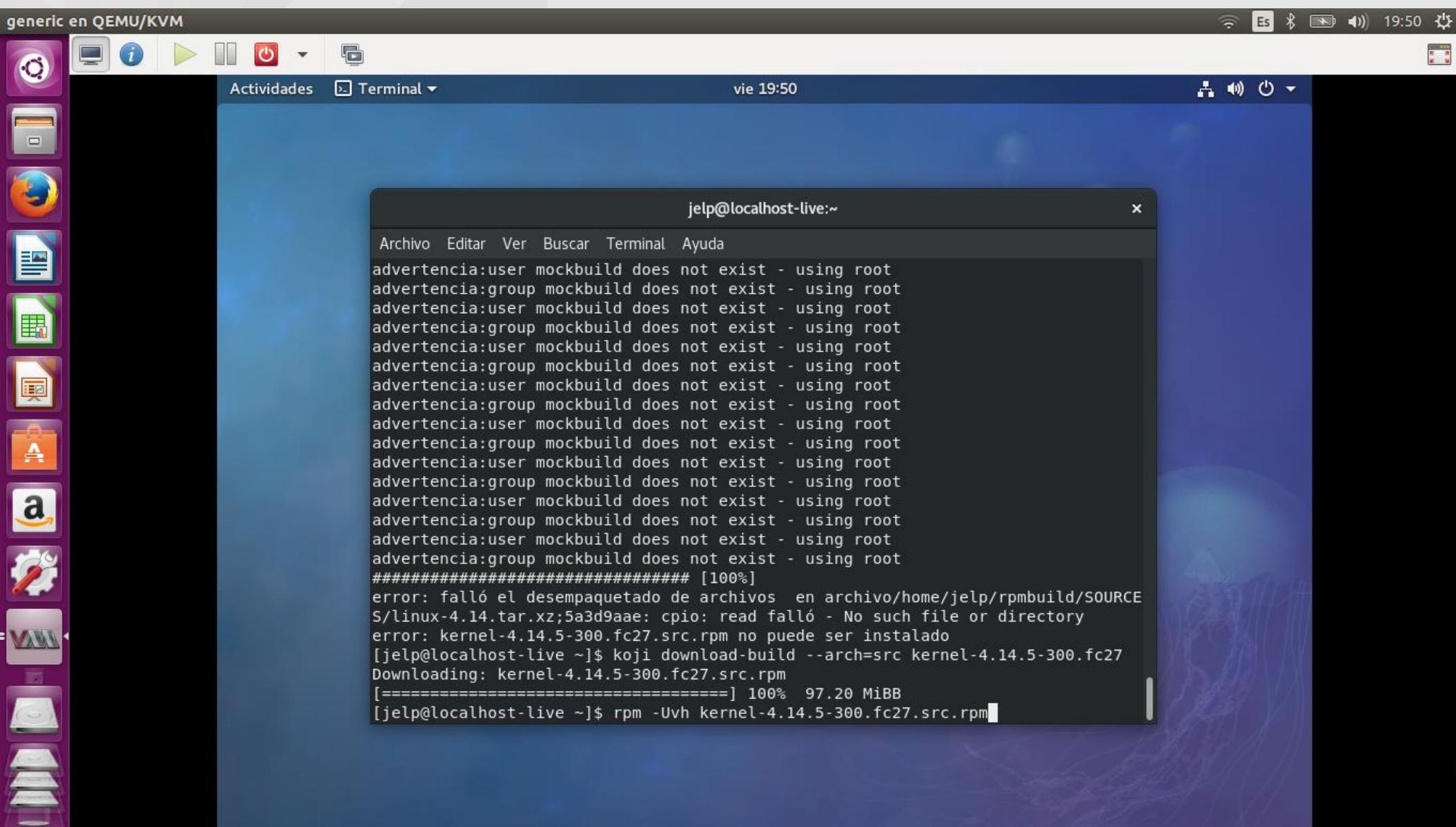
Por ejemplo



Ejecutamos el archivo rpm descargado:
Rpm -Uvh kernel-4.14.7.300.fc27.src.rpm



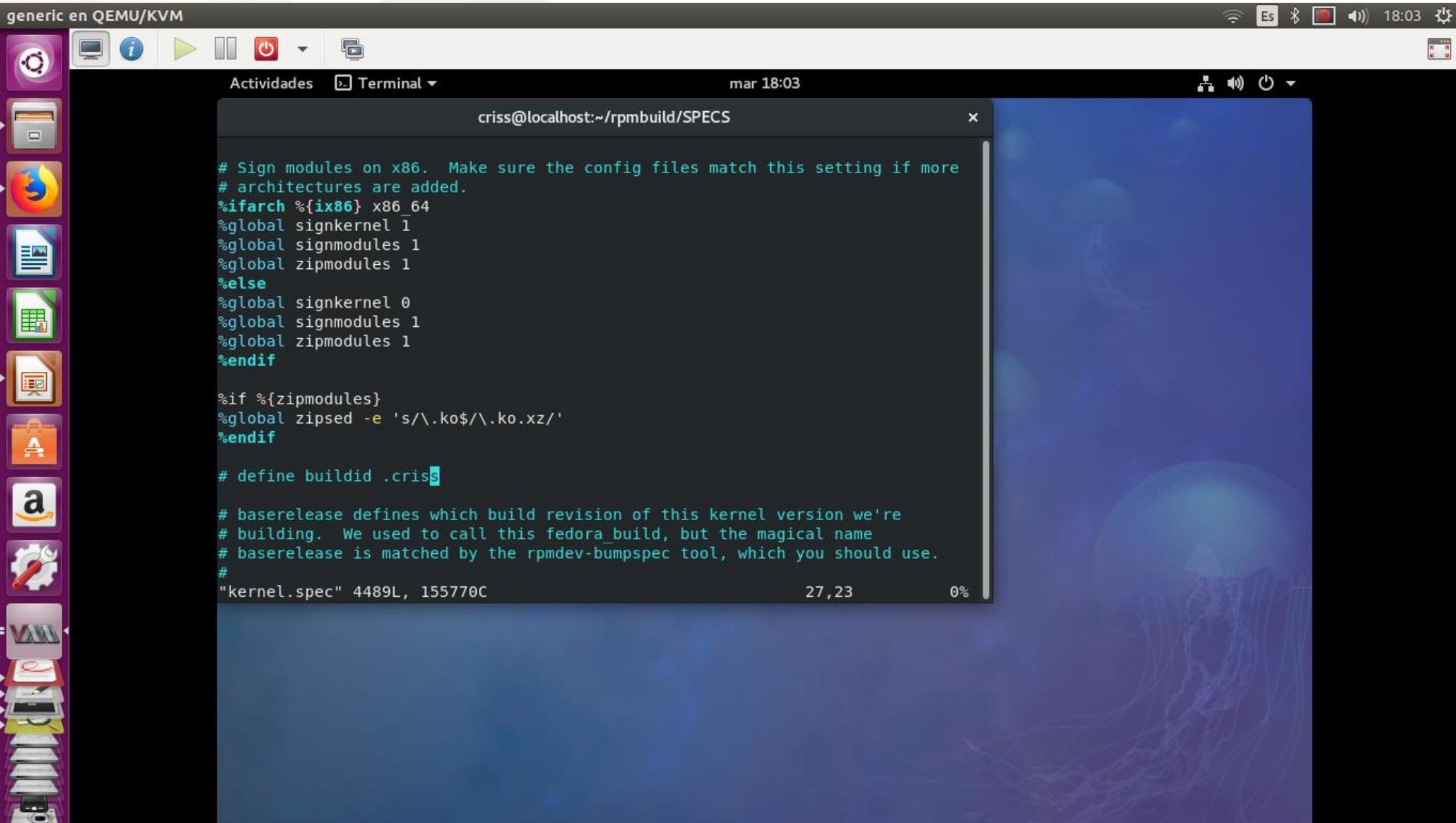
Ejecutamos el archivo rpm descargado: Rpm -Uvh kernel-4.14.5.300.fc27.src.rpm



Nos ubicamos en ~/rpmbuild/SPECS/kernel.spec

Modificamos el archivo kernel.spec con vi kernel.spec

Cambiamos #define buildid .local por #define buildid.<nombre de usuario>



The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "criss@localhost:~/rpmbuild/SPECS". The terminal content displays a portion of a kernel.spec file. The file includes configuration for module signing, conditional compilation based on architecture, and a section defining the build ID. A specific line "# define buildid .criss" is highlighted with a blue selection bar. The desktop background is a blue abstract image, and the taskbar at the top shows various application icons.

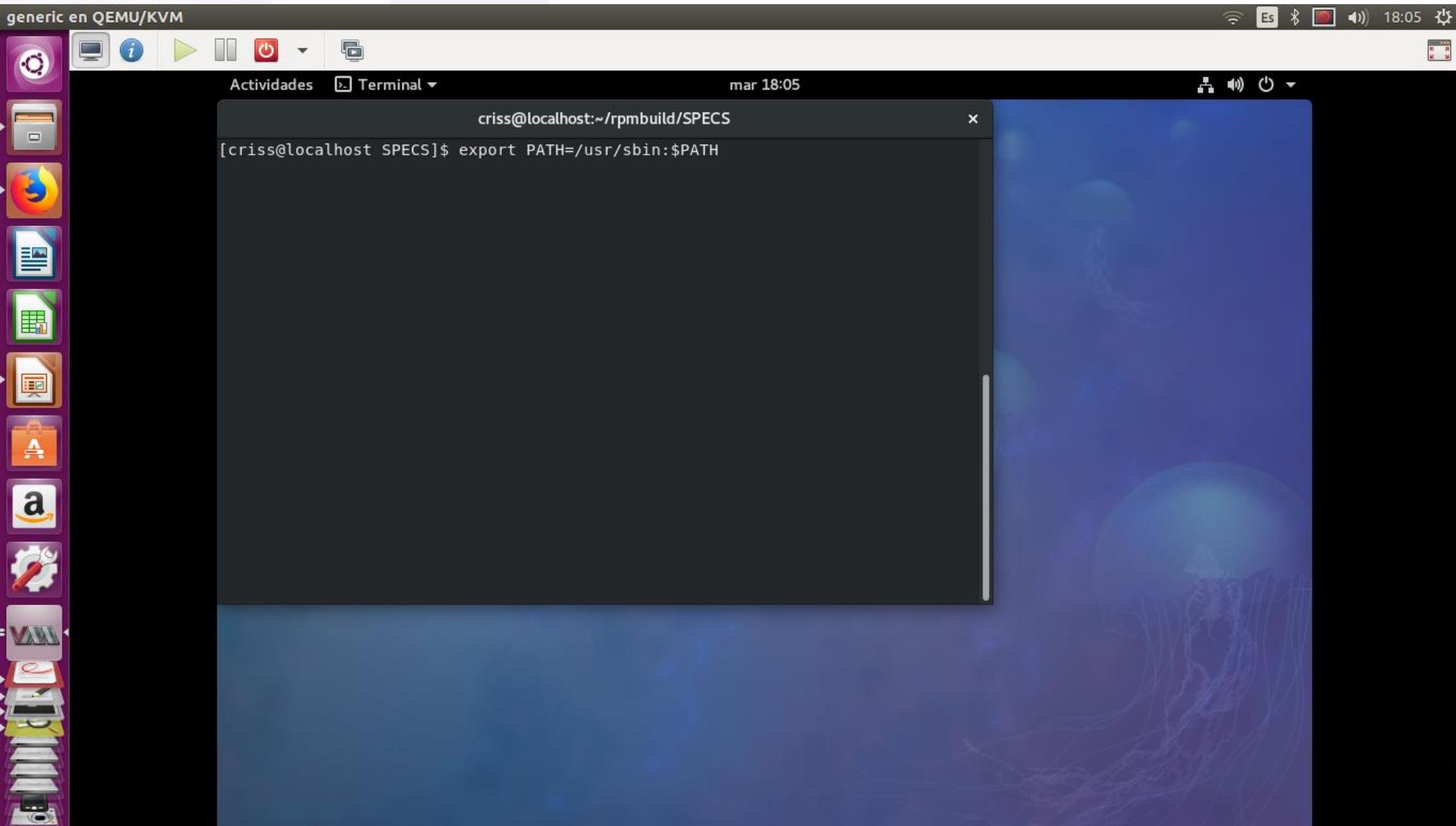
```
# Sign modules on x86. Make sure the config files match this setting if more
# architectures are added.
%ifarch %{ix86} x86_64
%global signkernel 1
%global signmodules 1
%global zipmodules 1
%else
%global signkernel 0
%global signmodules 1
%global zipmodules 1
%endif

%if %{zipmodules}
%global zipsed -e 's/\.ko$/.ko.xz/'
%endif

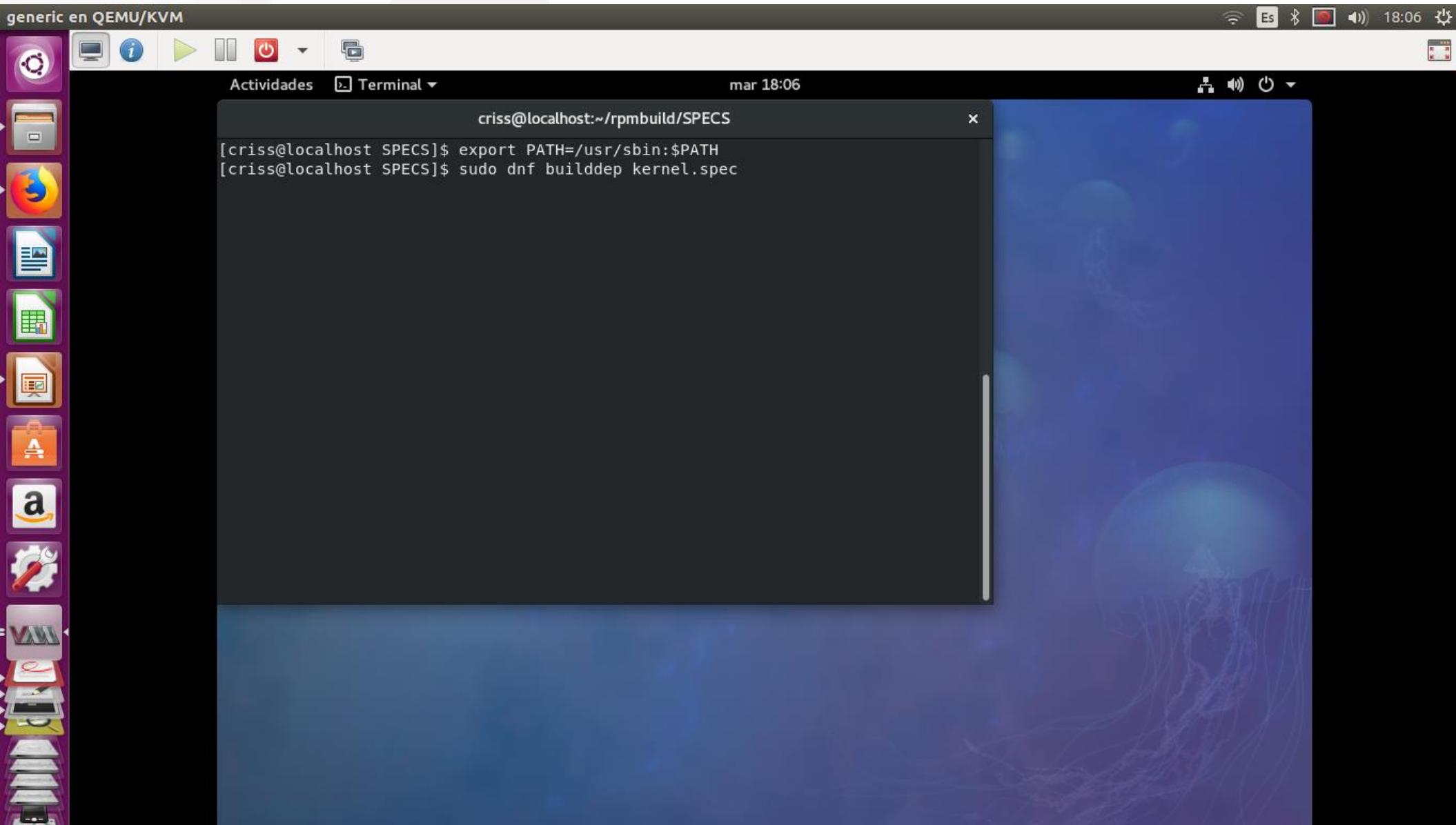
# define buildid .criss

# baserelease defines which build revision of this kernel version we're
# building. We used to call this fedora_build, but the magical name
# baserelease is matched by the rpmdev-bumpspec tool, which you should use.
#
"kernel.spec" 4489L, 155770C
```

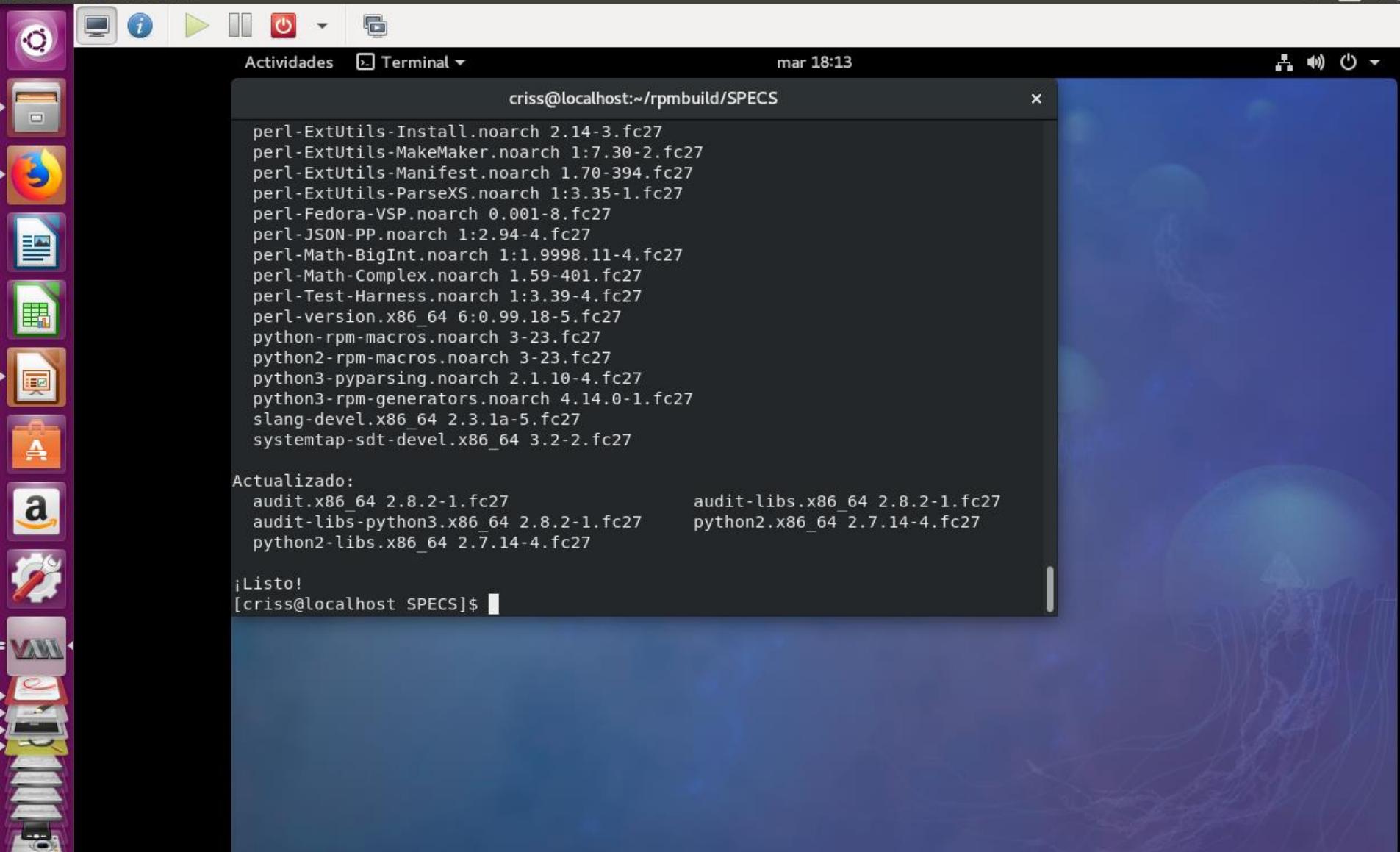
Nos ubicamos en ~/rpmbuild/SPECS/
Y ejecutamos la instrucción export PATH=/usr/sbin=\$PATH



Nos ubicamos en ~/rpmbuild/SPECS/kernel.spec
Luego ejecutamos el archivo de especificaciones de la siguiente manera:
Sdo dnf builddep kernel.spec

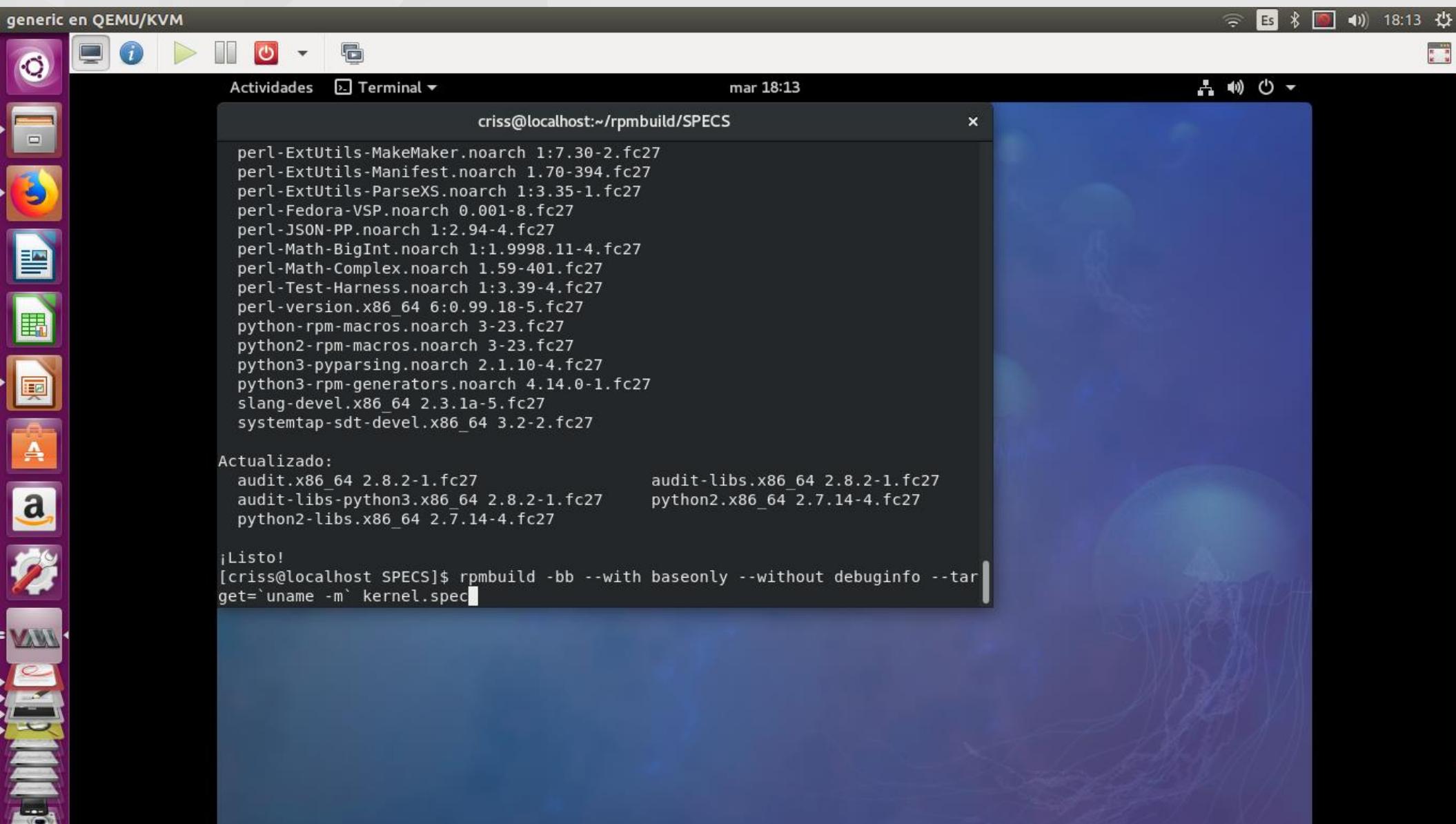






Nos ubicamos en ~/rpmbuild/SPECS/ Para compilar el kernel ejecutamos:

Rpmbuild -bb --with baseonly --without debuginfo --target='uname -m' kernel.spec

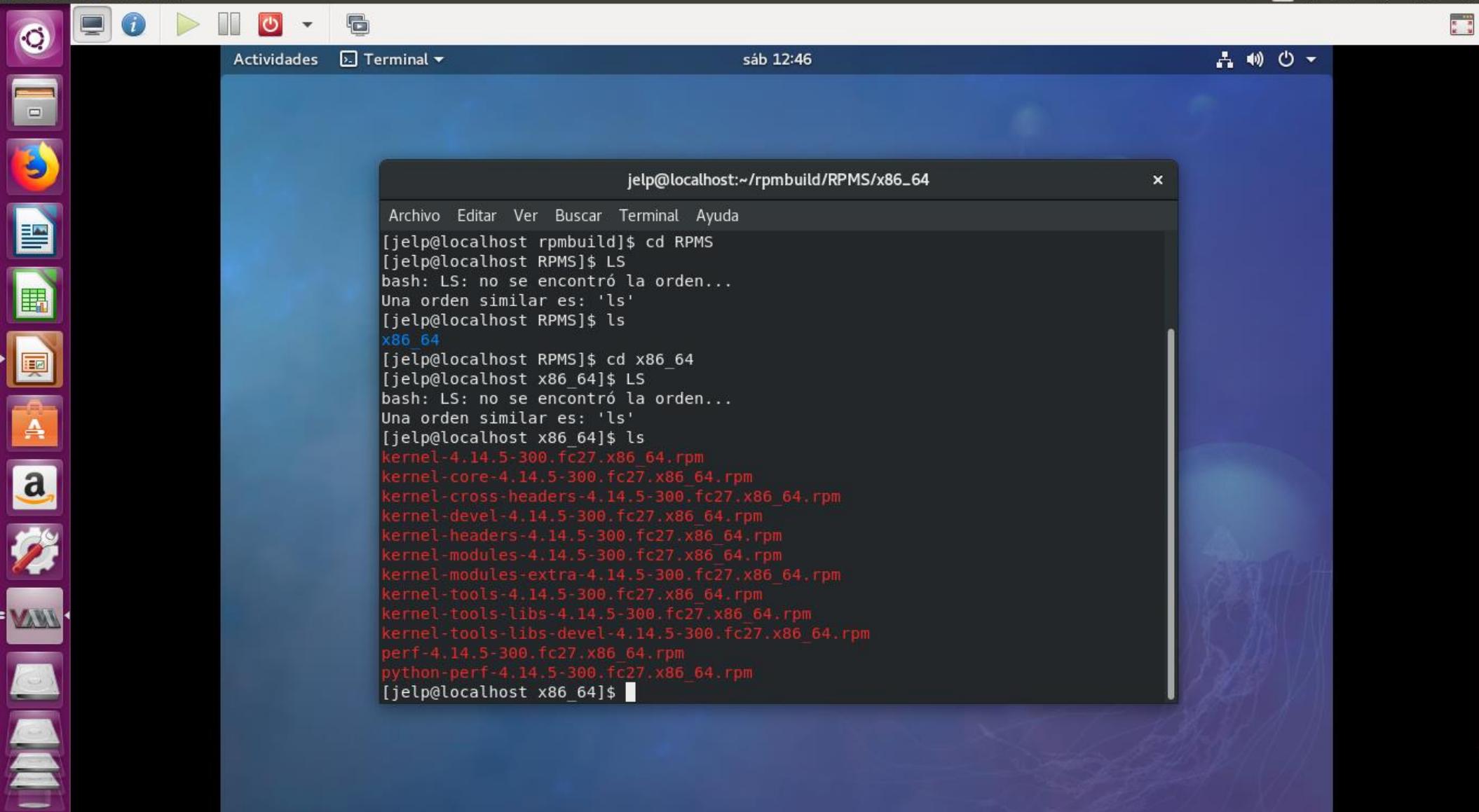


The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "Actividades Terminal" and the date and time are "mar 18:13". The terminal content displays the output of the rpmbuild command:

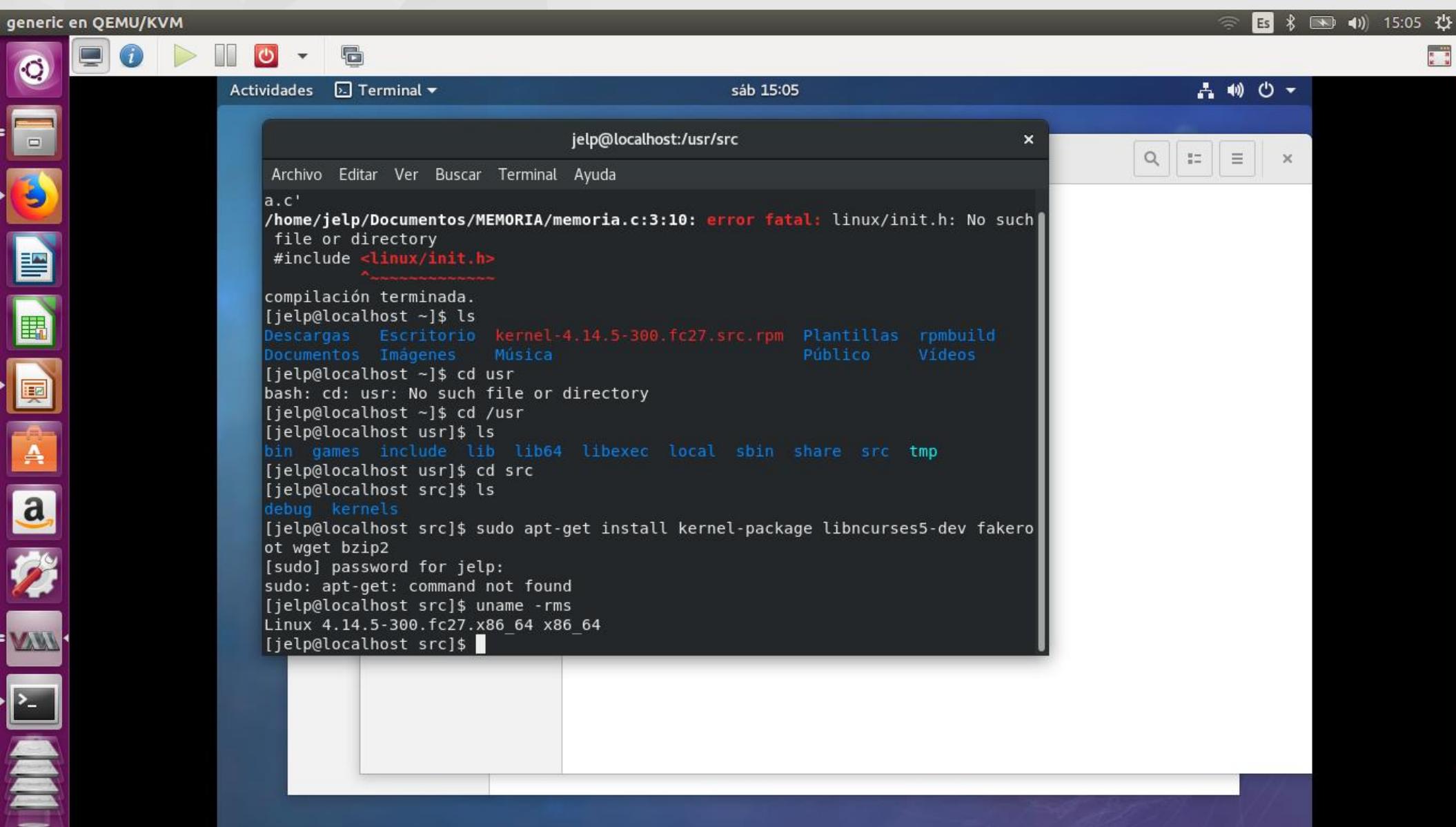
```
criss@localhost:~/rpmbuild/SPECS
perl-ExtUtils-MakeMaker.noarch 1:7.30-2.fc27
perl-ExtUtils-Manifest.noarch 1.70-394.fc27
perl-ExtUtils-ParseXS.noarch 1:3.35-1.fc27
perl-Fedora-VSP.noarch 0.001-8.fc27
perl-JSON-PP.noarch 1:2.94-4.fc27
perl-Math-BigInt.noarch 1:1.9998.11-4.fc27
perl-Math-Complex.noarch 1.59-401.fc27
perl-Test-Harness.noarch 1:3.39-4.fc27
perl-version.x86_64 6:0.99.18-5.fc27
python-rpm-macros.noarch 3-23.fc27
python2-rpm-macros.noarch 3-23.fc27
python3-pyparsing.noarch 2.1.10-4.fc27
python3-rpm-generators.noarch 4.14.0-1.fc27
slang-devel.x86_64 2.3.1a-5.fc27
systemtap-sdt-devel.x86_64 3.2-2.fc27

Actualizado:
audit.x86_64 2.8.2-1.fc27          audit-libs.x86_64 2.8.2-1.fc27
audit-libs-python3.x86_64 2.8.2-1.fc27    python2.x86_64 2.7.14-4.fc27
python2-libs.x86_64 2.7.14-4.fc27

¡Listo!
[criss@localhost SPECS]$ rpmbuild -bb --with baseonly --without debuginfo --tar
get=`uname -m` kernel.spec
```



Una vez terminado el proceso de compilación utilizamos el comando `uname -rms`



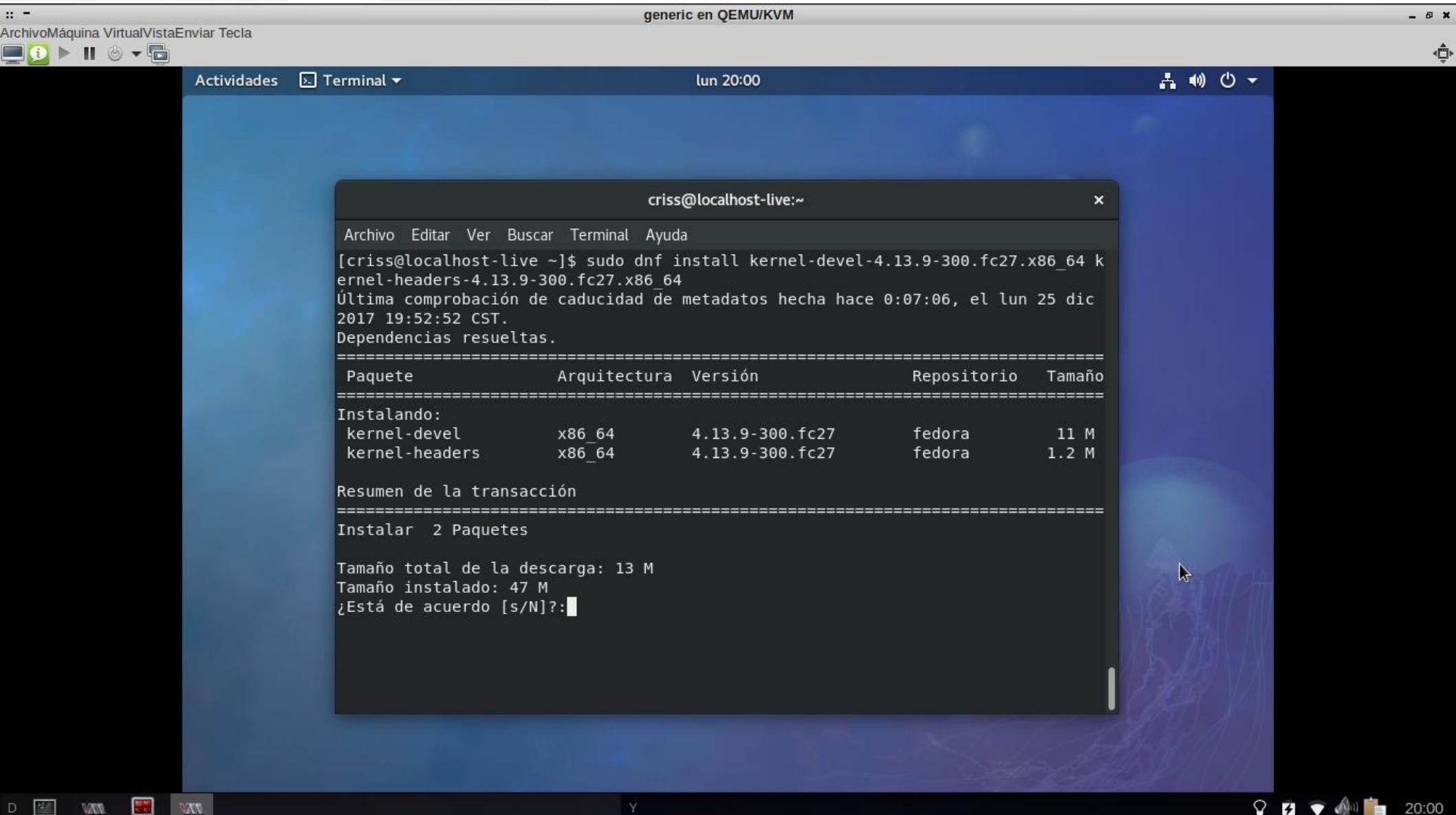
COMPILEACIÓN DE MODULOS KERNEL EN FEDORA 27

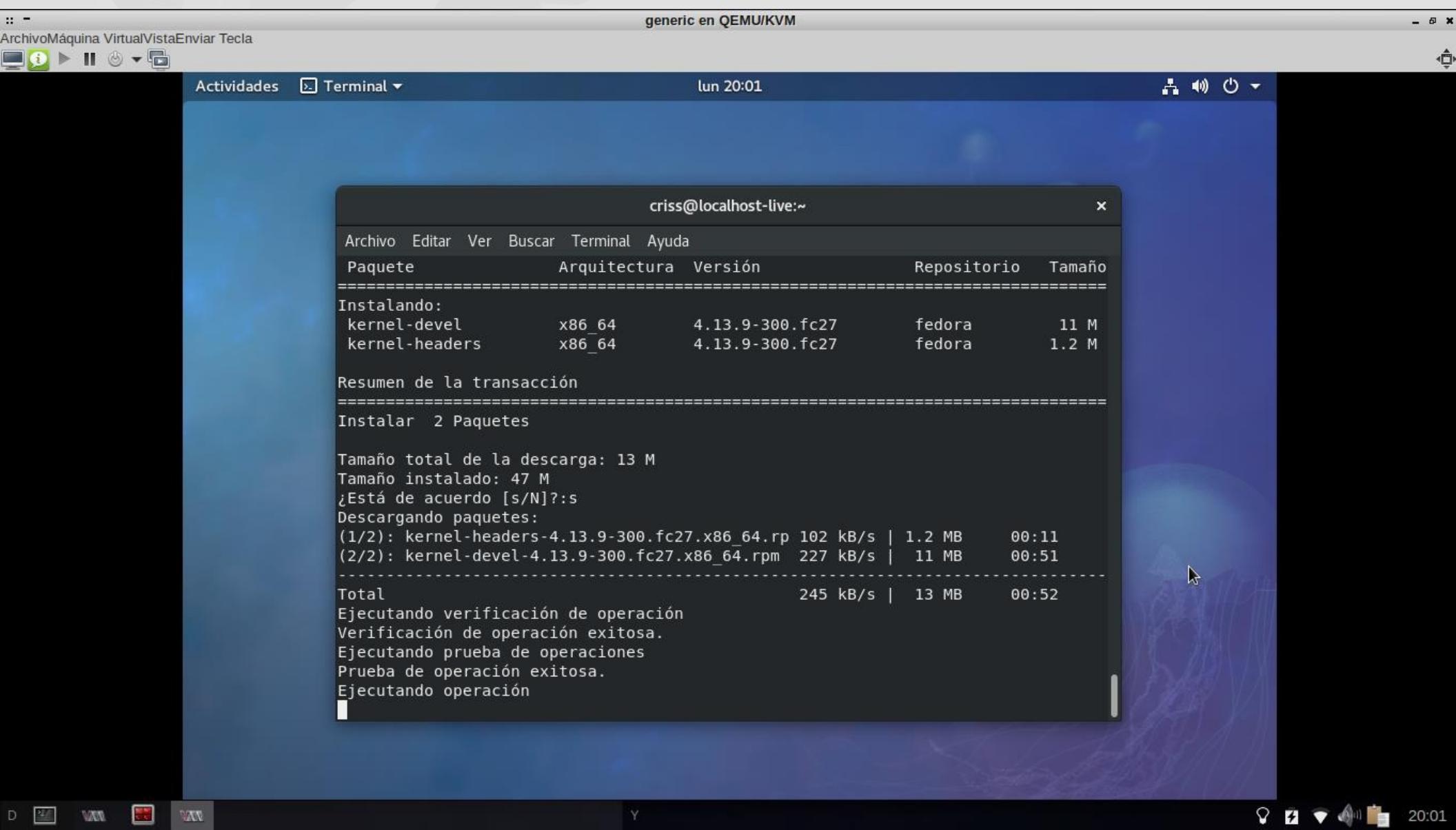
Instalación de herramientas y preparación de ambiente para compilación

(kernel 4.13.9-300)

Instalamos las herramientas necesarias:

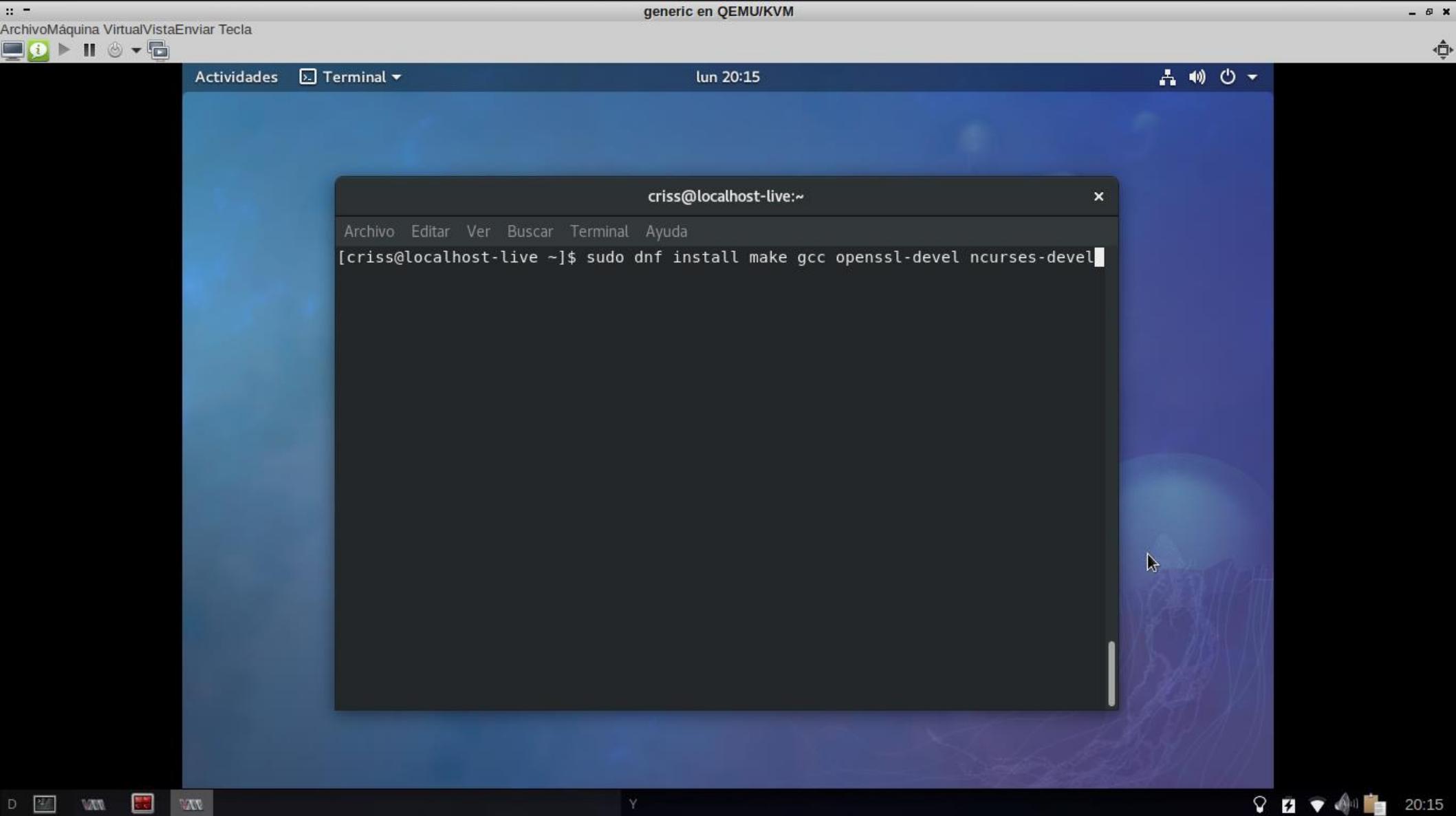
```
dnf install kernel-devel-4.13.9-300.fc27.x86_64 kernel-headers-4.13.9-300.fc27.x86_64
```

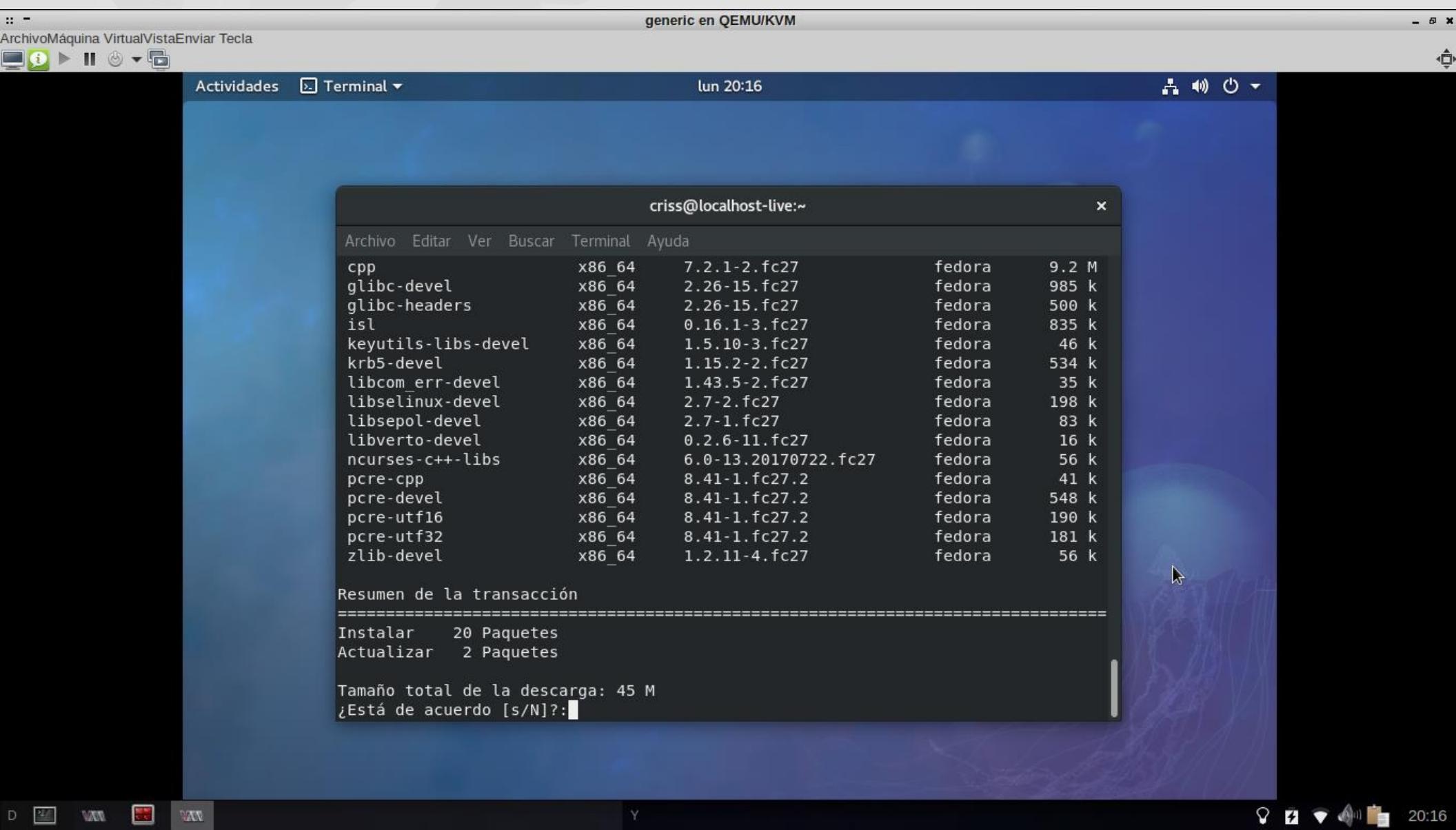




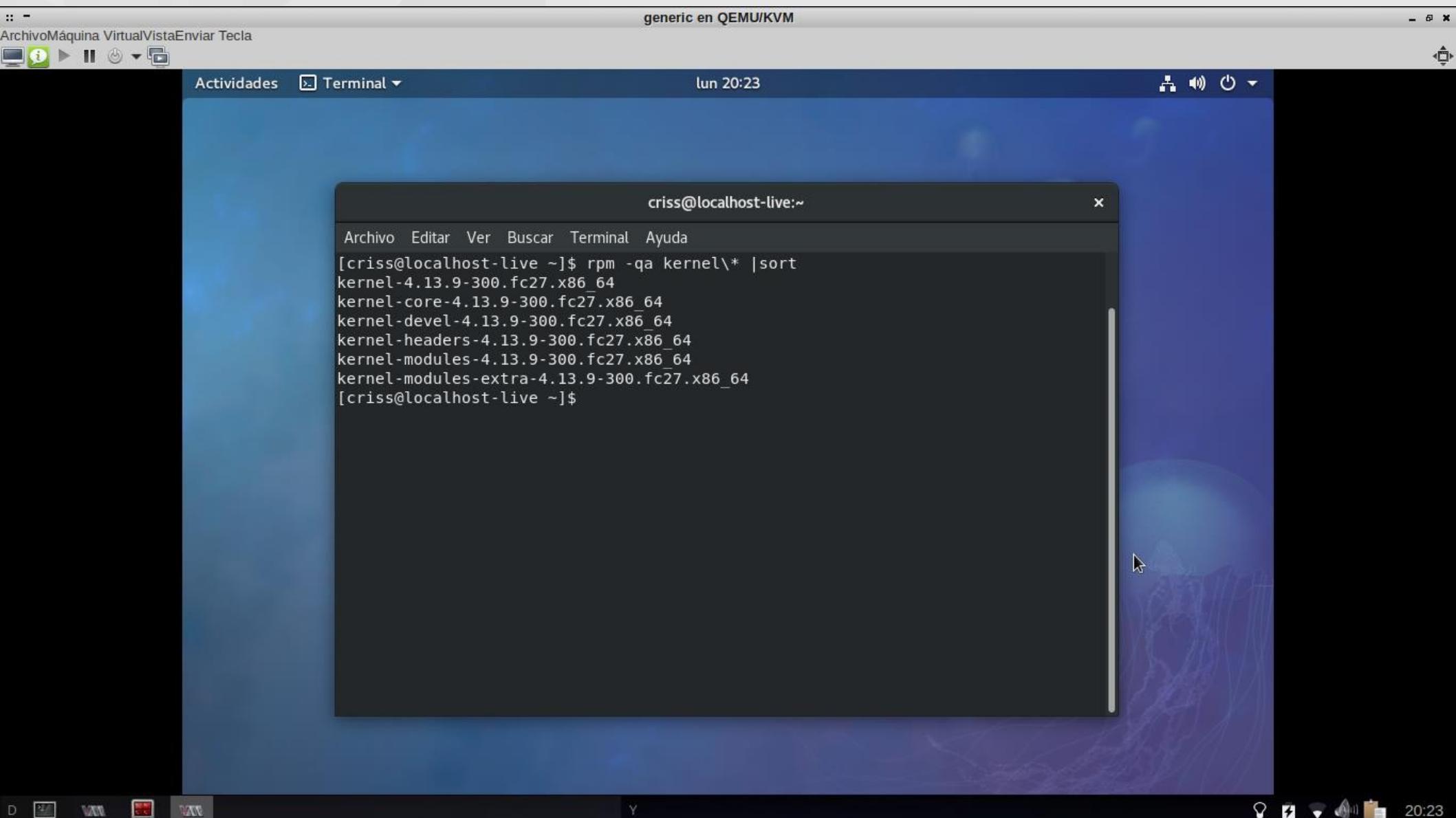
Instalamos otras librerias para poder ejecutar el Makefile del modulo.

Dnf install make gcc openssl-devel ncurses-devel





Kernel 4.13.9-300



COMPILACIÓN DE MODULOS KERNEL EN FEDORA 27

Compilación de modulos
(kernel 4.13.9-300)

Para poder compilar un modulo se necesita lo siguiente:
modulo.c

Makefile

Donde “modulo” es el nombre del archivo en lenguaje C que contiene las instrucciones.

Y Makefile es el archivo que se ejecutara tras la instrucción en consola “make”.

Situados en el directorio en donde tenemos estos archivos ejecutamos la sentencia:

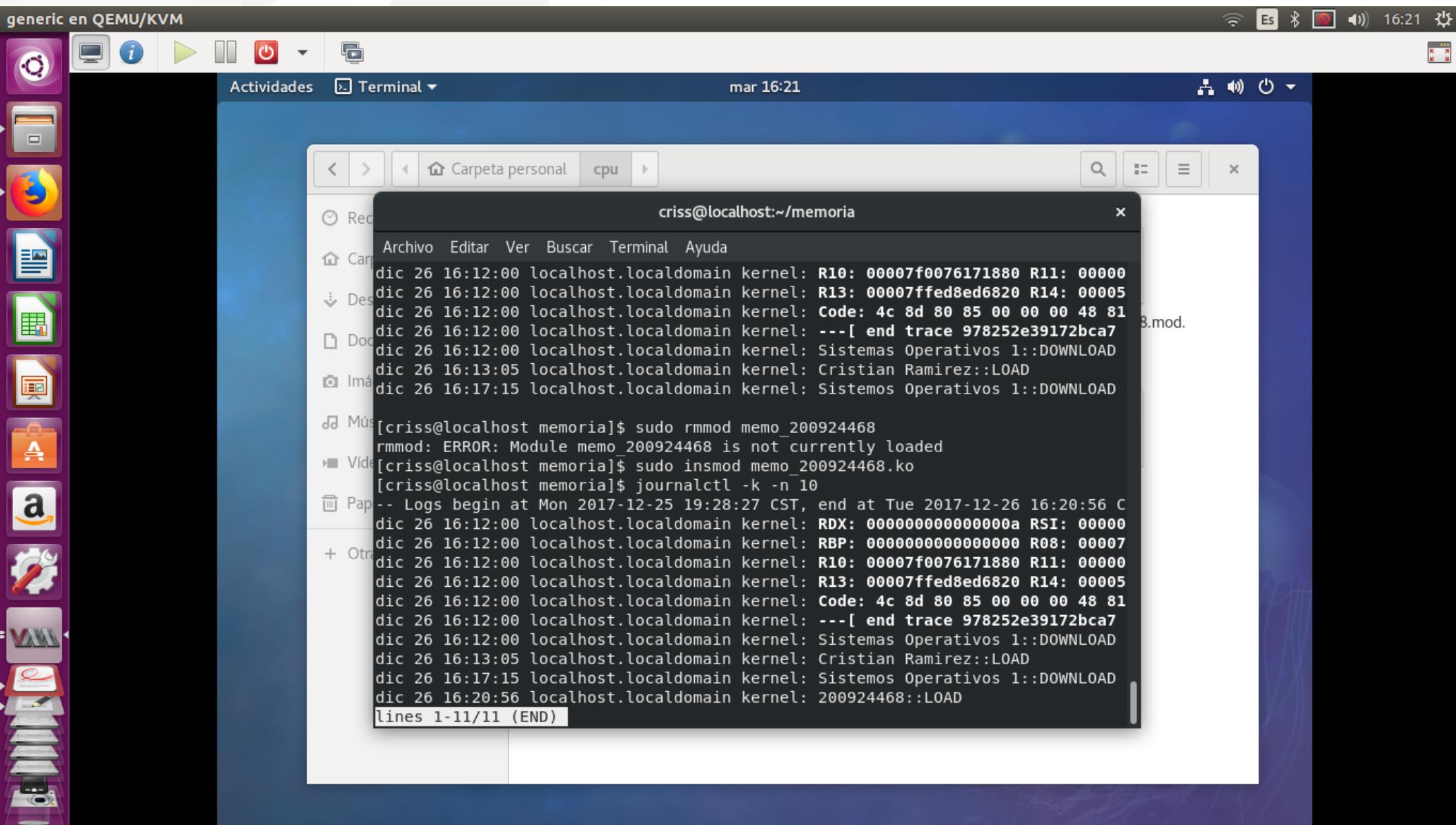
make

Esto generará el modulo.o requerido para la compilación en Makefile, a la vez que Makefile generará el modulo.ko, que con las instrucción “**sudo insmod modulo.ko**” podremos cargar la kernel.

Para removerlo utilizamos “**sudo rmmod modulo**”.

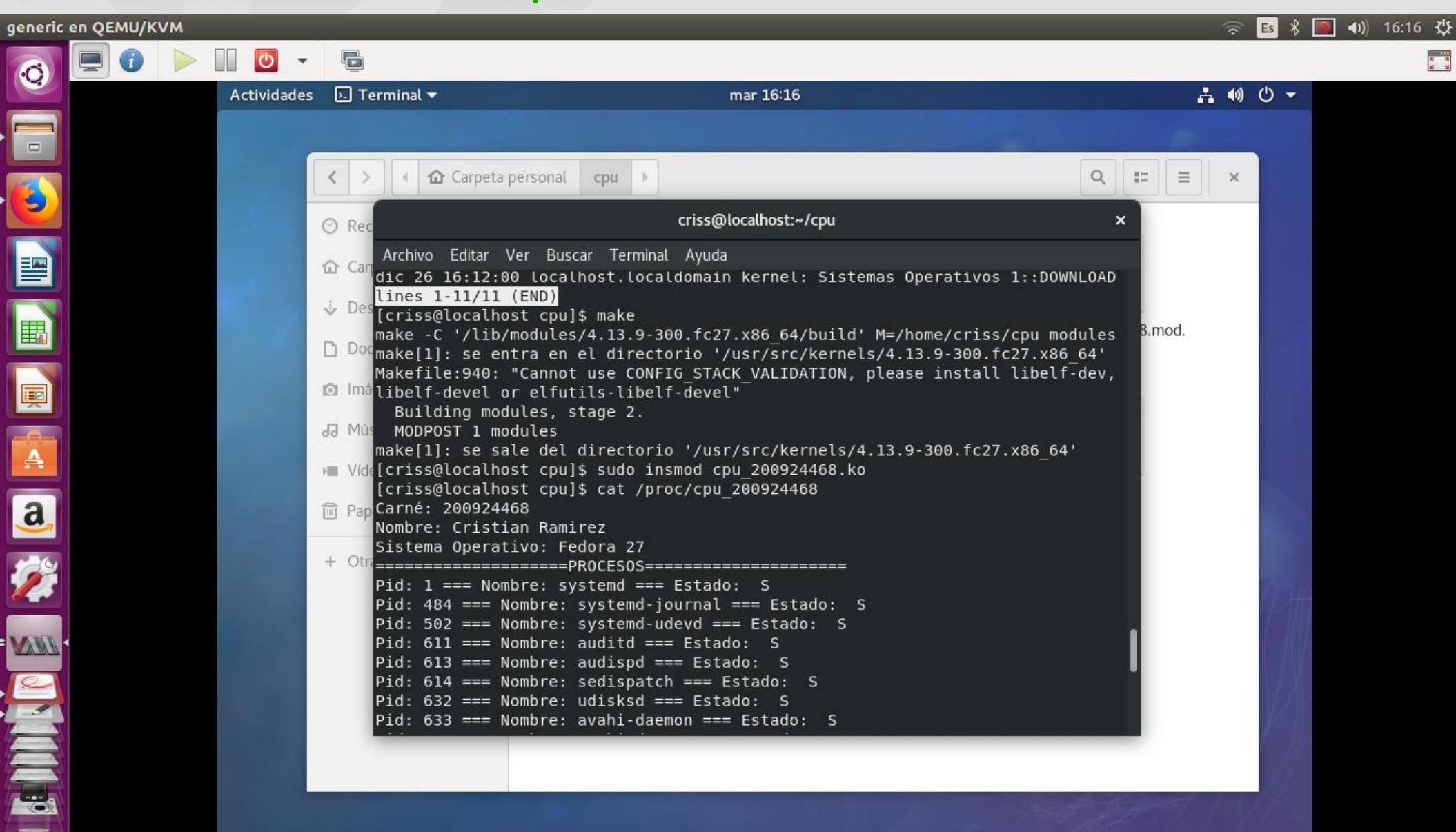
Estas operaciones Podemos verlas con “**journalctl**”.

Utilizando “journalctl” para ver las operaciones sobre el modulo.



Para poder ver la “acción” definida para el modulo que acabamos de agregar utilizamos:
`cat ~proc/<nombre_archivo>`

Cpu_200924468



Asignar el tamaño de RAM a utilizar junto con la cantidad de núcleos que desea asignarse.