Hibernation

Hibernation (also known as suspend-to-disk) is a method used to power down a computer while preserving the state of the operating system. Once powered up, the operating system will be in the exact state as it was at the time of hibernation.

In Slackware, hibernation works out-of-the-box. However, in order to resume successfully from a hibernated state, you need to configure the bootloader so it knows where to locate the resume data.

Hibernating

Desktop Environments

KDE, XFCE and other desktop environments offer hibernation from logout screen/menu.

Lightweight Window Managers

You can hibernate your system from console. Either as root via pm-utils or D-Bus with additional programs as regular user.

pm-utils

Hibernation is done via `pm-hibernate` command.

D-Bus

Slackware offers two services to control power management in it's recent releases. UPower and HAL. Since HAL is deprecated in Slackware 14.0, use HAL in older Slackware releases (from 12.0 to 13.37), otherwise use UPower.

Use one of these command to hibernate your system.

UPower

```
 $ dbus-send --system --print-reply --dest="org.freedesktop.UPower" \
 /org/freedesktop/UPower org.freedesktop.UPower.Hibernate
```

HAL

```
$ dbus-send --system --print-reply --dest="org.freedesktop.Hal" \
 /org/freedesktop/Hal/devices/computer
org.freedesktop.Hal.Device.SystemPowerManagement.Hibernate
```

Bootloader configuration

Booting from LVM

LILO doesn't support booting from LVM, therefore you have to create initial ramdisk (initrd). You need one even though your kernel has build-in support for LVM.

You don't have to specially adjust `lilo.conf` because you can set everything resume-related in initrd.

```
mkinitrd -c -k <kernel-version> -f <fs_type> -m <modules_required_to_boot> -
r <root_partition> -h /dev/volumegroup/swap -L
```

- -h set your hibernation/swap logical volume
- -L activate LVM

LILO configuration

Next step is to make sure your `lilo.conf` contains following snippet

```
image = /boot/vmlinuz
  root = /dev/sdaX
  initrd = /boot/initrd.gz
  label = Slackware
  read-only # Partitions should be mounted read-only
```

If you don't use LVM, you can simply tell kernel where to resume from by adding append line:

```
append="resume=/dev/sdaX"
```

Replace `/dev/sdaX` with your actual swap partition.

Afterwards, run `lilo` to save changes.

```
$ lilo
```

Lock your computer upon resuming

You may have noticed that when your computer resumes, there is no protection to ensure no one can use your computer. As `pm-hibernate` must be run as root, any user that powers up your computer will be granted a root terminal! I'll show you some tricks to ensure a proper lock of the system.

Security warning: whether you are using the trick from the command-line or from an X session, don't forget to logout from all other ttys! None of them will get locked, so people issuing CTRL+ALT+F1 (ar any Fx key) upon resuming will be able to access them, rendering the lock trick totally useless!

With pm-utils

From the command line

If you dont have an X session open, the only way to lock your computer is to logout from your session before hibernation. You can use the following command:

```
pm-hibernate & logout (OR pm-hibernate & exit)
```

It's important to logout from every tty prior to hibernation! `logout` will only log you out from the terminal from where you launched the command. Please read the warning in the following section.

This trick may not work on fast systems, as `logout` may not have enough time to complete. You may think that using `;` instead of & could do the trick (as it will log out the user *after* resuming), but it allows the user doing a CTRL+Ctrl upon resuming, killing the `logout` command.

From an X session

If you want to lock the computer prior hibernation, you can issue the following command (as root of course):

```
# xlock & sleep 5 ; pm-hibernate
```

You can also create the following script:

```
#!/bin/bash
echo 'Hibernating in 5 seconds'
xlock &
sleep 5
pm-hibernate
```

`xlock` is a command that locks your X session. Upon rebooting, you will be asked for the *root* password. The `sleep` command may be omitted, but I think it's safer to leave it, so you are sure that `xlock` has enough time to run properly. You don't have to use `exit` (or `logout`) at the end of the command or the script, as it has no use concerning security. You may want to use it though to automatically close the xterm on resuming.

Security warning: this “trick” will only lock the X system! If you are logged on as root in any of the other ttys, `xlock` can be killed, leaving full access to the X session.
For people using init3 (no automatic X startup), remember to also logout from the tty from where you launched the X system with `startx`: hit CTRL+Z (to suspend the task), then bg (to resume the task in the background), then CTRL+D (or `logout`) to logout.

An alternate

If one is a sudoer, ie, can use sudo, then the following can also be used:

```
sudo /usr/sbin/pm-hibernate
```

Further reading

- Hibernate to encrypted swap - README_CRYPT.TXT

Sources

- Original source: http://www.slackwiki.com/Hibernate
  - Original author unknown - original submissions 1,2
  - Otherwise rewritten by Martin Matějek (also on Slackwiki)
  - Hibernate trick by zithro

howtos, hardware, hibernation, power saving, author yenn, author zithro

From:
https://docs.slackware.com/ - **SlackDocs**

Permanent link:
https://docs.slackware.com/howtos:slackware_admin:hibernation

Last update: **2014/01/10 05:51 (UTC)**