# Linux Swap Space Mini−HOWTO

# Table of Contents

# Linux Swap Space Mini−HOWTO


## Rahul U. Joshi,
## jurahul@hotmail.com http://jurahul.freeservers.com


v1.42, 18 January 2000

---

*This Linux Mini−HOWTO describes how to share your Linux swap partition with Windows.*

---

# 11. The msinfo.sh script

# 12. The original msinfo file

# 13. Acknowledgements, feedback and dedication

---

# 1. Introduction

Many people use both Linux and MS−Windows. The ability to do so is an important part of "the Linux revolution"; i.e. letting people experiment with (and get hooked on) Linux while still being able to run their off−the−shelf software. Since both Linux and MS−Windows use virtual memory with swap to disk, a frequently occurring question in comp.os.linux.setup is how to share swap spaces, in order to reduce the amount of disk space needed.

There are several methods for sharing swap spaces, the one described in this document is probably the most complicated one but is the only one I have encountered that allows maximum performance for both environments without the risk of trashing a disk partition. If you are using DOS, then many DOS applications manage their own swap files. You can also share the Linux swap partition with these applications.

---

# 2. What you need

This procedure have a few requirements that need to be filled. I strongly recommend that you fill these requirements *anyway*, as there are several problems with older versions.

1. MS−DOS 5.0 or newer and MS−Windows 3.1/3.11 **OR** MS−Windows 95/98
2. A shutdown/init that knows to run a file on shutdown. (The SysVinit−2.50 package can do this, for example. SysVinit−2.50 is available from sunsite.unc.edu in /pub/Linux/system/Daemons. Almost all current distributions use this init package.)

---

# 3. Recent Changes and versions

- Version 1.3 −− Written by H. Peter Anvin, and was not maintained from 15 Jan 1998.
- Version 1.4 −− Based on v1.3, but added more details. Currently maintained by Rahul U. Joshi
- Version 1.42 −− Deleted the recovery section for v1.4 users an v1.4 never actually appeared on the LDP website.

Modifications by Rahul U. Joshi

Date: 26 Nov 1999

- Added Windows 95/98 specific details.
- Added Red Hat Linux specific details.
- Modified the msinfo program.
- Added sections 5 to 12.

- Completely revamped the "Procedure" section to make it more understandable.
- Split up the swap initialization and halt into two shell scripts.
- Modified the scripts to handle signature verification more than once.
- Added details about new type swap areas.
- Added more reliable schemes.

# 4. Administrative

## 4.1 Copyright



## 4.2 Disclaimer

Although I have tried my best to bring out various facts, the author is not responsible for any damage due to the actions taken based upon the information contained in this document. It is impossible to test the things under all the configurations, so probably some of the hints given in this document may be buggy and may not work on your system. In case you trace the bug, let me know it first !

This document is provided ``as is''. I put great effort into writing it as accurately as I could, but you use the information contained in it at your own risk. In no event shall I be liable for any damages resulting from the use of this work.

In case you are planning to include this HOWTO on some distribution medium or in print, I will like to have an acknowledgement e−mail (just for my record). In that case, I may also be able to send you the latest version of the document.

# 5. Before you begin

This is **not** a step by step HOWTO. The reader is expected to first read the entire document so that he (or she) becomes aware of the possible pitfalls and performance bottlenecks. After having a fair idea as to what steps are to be taken, he (or she) can then follow the next section in a step by step manner. Since you will be playing as a root and that too with the system initialization files, be alert as to what you are doing. Write in those files comments for those sections that you added, and when you want to remove a section, do not delete it. Instead just comment it out. (In shell scripts comments begin with a pound(#) sign). In case things go wrong, you may restore your original configuration by removing the changes you did. Although you are not required to know shell scripting since I have given all those that would be needed, some knowledge may be advantageous both for understanding as well for debugging.

---

# 6. The Procedure

*NOTE:* This procedure has been written keeping in mind RedHat Linux 6.0 Although this procedure in general is applicable for all Linux distributions, the details may vary. You are welcome to add the details for your distribution. Many users will already have a swap partition devoted to Linux. I assume you have one.

## 6.1 Turn off swapping and create a DOS partition

- Boot Linux. Turn off swapping by editing the /etc/fstab file and commenting the line that describes your swap partition.
- Using fdisk under Linux, delete the swap partition in order to create free space on the disk.
- Boot DOS. Create a DOS partition (using FDISK) the size = the size swap space you want. If you cannot create the partition (probably because DOS FDISK says that all the space in extended partition is allocated to the logical drive), you can use fdisk or cfdisk under Linux to create the partition. You can even just change the type of your swap partition from 82h (Linux Swap) to 06h (FAT16). In that case you don't need to delete the swap partition, just change the type.
- Once you have created the partition, Boot DOS if you are not already in DOS. DOS will be assigned a drive letter to your new partition. Use that drive letter instead of X whenever these instructions lists a command like "LABEL X:" or "COPY FOO X:DUMMY.DAT"
- Format this partition using the DOS FORMAT command.

  ```
  C:\>FORMAT X:
  ```
- Set the volume label on this partition to "SWAP SPACE" using the DOS LABEL command. Verify it by the DIR command. Please do this as a separate step. Some versions of FORMAT do not seem to put the volume label in the boot sector as it should. [Note: some people has written me saying the volume label is stored in the root directory. Yes, but at least since DOS 5.0 it has also been in the boot sector.]

  ```
  C:\>LABEL X:
  C:\>DIR X:
  ```
-

At this stage, you have a DOS partition ready for use for swapping.

## 6.2 Tell Windows the location of the new swap file

### For Windows 3.1 users

- Start Windows. Go to the Control Panel, select "386 Enhanced". Select "Virtual Memory" and create a Windows Permanent swap file on drive X: of maximum size (Windows will tell you the maximum size). Windows may complain saying it will not use a swap file that big. Ignore the message and create the file anyway.
- Exit Windows.

### For Windows 95/98 users

- Start Windows. Go to the Control Panel (Start | Settings | Control panel). Double click the System icon. In the resulting System Properties dialog box, select the Performance tab. Now click the Virtual Memory button. This will bring up the Virtual Memory dialog box. Select "Let me specify my own virtual memory settings". Select the drive you intend to share as swap space with Linux (X:). Keep the Maximum and Minimum values as selected by Windows. Click OK. Windows will prompt you to restart your computer. Do it.
- When you restart Windows, you will see that the win386.swp file has been created on the X: drive. In addition, there is also the Recycled folder. Since you don't intend to store any other files on this drive and also a single file on this partition will be convienent, tell Windows not to maintain Recycle Bin for this (X:) drive. On the Desktop, left click the Recycle Bin icon, and in the resulting pop−up menu, click Properties. This will bring up the Recycle Bin Properties dialog box. Click the Global tab and select "Configure drives independently". Now click the tab for X: drive (which will have the label SWAP SPACE). In this dialog box, move the slider for "Maximum size of Recycle Bin" to 0 percent. The "Space Reserved" entry should now read as 0 bytes. Click OK. Restart your computer.
- Now that you are not maintaining the Recycle Bin for drive X:, ideally the `RECYCLED` folder from X: must be deleted by Windows. Not only Windows doesn't does that, but also does not allow you to delete the folder. But as I said earlier, this folder is now unnecessary and we want drive X: to contain only the win386.swp file. For that purpose, first start your computer in MS−DOS mode. Select Start | Shut Down | Restart your computer in MS−DOS mode, or if you are booting Windows, press F8 at the start, and then from the startup menu, select Command Prompt only. Once you are in MS−DOS mode, type the following commands:

  ```
  C:\>X:
  X:\>ATTRIB −R −H −S RECYCLED
  X:\>DELTREE RECYCLED
  ```

  Now you have drive X: containing only win386.swp. You can verify this using the various options for the `dir`command. This does not mean that the `RECYCLED` folder will not be there on drive X:.

When Windows boots up and finds that there is no `RECYCLED` folder on X:, it will create it automatically. However, when we will be restoring X: from Linux, it will simply contain win386.swp file. If anyone is aware of a method to prevent Windows from creating the `RECYCLED` folder, please e−mail me.

- **Disk Cleanup troubles you ?**

  If you have Disk Cleanup installed, then ocassionally Windows comes out with a "Low disk space on drive X:". Since we are using the entire X: drive for swapping, there is no need to have any free space on X:. To prevent this dialog box from appearing, start Disk Cleanup (point to Start | Programs | Accessories | System Tools | Disk Cleanup). In the resulting dialog box, select the X: drive and click OK. The "Disk Cleanup for drive X:" dialog box appears. Click the "Settings" tab and uncheck the "If this drive runs low on disk ... blah blah blah" check box.

>From this stage onwards, Windows will assume that it's swap file is on drive X:. So the drive X: must be intact each time you boot Windows. If you are using some additional system utilities like Norton Utilities for Windows 95, then you should probably consult the online help or the manuals to keep them informed of the changes in the settings, otherwise they may come out with an error message.

## 6.3 Back up the Total Special Sectors

- Boot Linux, then log in as root.

- Use the fdisk command to find the name of the partition and its size in blocks. Create a symbolic link from `/dev/winswap` to this partition. If the partition is hda7, then type:

  ```
  # ln −s /dev/hda7 /dev/winswap
  ```

  [NOTE TO PURISTS: Please use a symlink. The name of this partition is going to go into several configuration files and inconsistencies could be fatal.]

- Find the "Total Special Sectors" for the swap drive. These are nothing but the total number of sectors required for the boot sector, FAT and root directory. There are a number of ways to find this number. Simplest copy the msinfo.sh file using your favourite editor. Give the following commands at the shell prompt

  ```
  # cp Swap−Space−HOWTO msinfo.sh
  # chmod +x msinfo.sh
  ```

  (Much easier if you use one of the editors with KDE or GNOME)

  Now, open the msinfo.sh file in the editor and delete everything except the msinfo.sh file in section 8. Now run this file as

  ```
  # ./msinfo.sh /dev/winswap
  ```

The program will print the some information about the swap partition. Take note at the number saying "Total special sectors", and verify that the volume label says "SWAP SPACE". If it does not, reboot DOS and re−do the LABEL command.

- **Note:** You can also find the "Total Special Sectors" using tools that can display the Boot sector like the Norton Disk Editor or minfo from mtools under Linux. For Norton Disk Editor, select "Boot Sector" from the Object menu. Note down the following entries :

  ```
  Reserved Sectors at beginning : r
  FAT Copies                    : f
  Sectors per FAT               : s
  Root directory entries        : r
  ```

  Now the "Total Special Sectors" is given by

  T = r + (s * f) + (d / 16)

  However, using the above program is more convienent.

  Let us denote these Total Special Sectors by XXX.

- [Optional step] Windows may occasionally leave some space on the partition, even if it is told not to. Don't attempt to use this space, since it will be erased any time you run Linux. If you want to avoid accidentally using it (and lose data), you can create a dummy file that fills that space by using the following commands:

  ```
  mkdir /mnt
  mount −t msdos /dev/winswap /mnt
  dd if=/dev/zero of=/mnt/dummy.fil
  umount /mnt
  ```

  The dd command will report "No space left on device". This is exactly what you want.

- Type on the shell prompt:

  ```
  # dd if=/dev/winswap bs=512 count=XXX | gzip −9 > /etc/winswap.gz
                                    ^^^
  ```

  ... where XXX is replaced with the "Total special sectors" number.

  Here we are saving the Total Special Sectors in a compressed form in the file /etc/winswap.gz

## 6.4 Modify the initialization and shutdown scripts to handle our new configuration

- Now, we will modify our initalization scripts so that swap space will be create on the partition each time Linux starts and the DOS/Windows special sectors will be restored each time Linux shut downs.

Instead of placing the command for handling the details directly in the initialization scripts, we will prepare two separate files called swapinit.sh and swaphalt.sh. One sample of these files that should work under Red Hat Linux is given in section at the end. You can choose to place these files under any directory, preferably /etc/rc.d/init.d. Create the two files in this directory and copy into then the scripts given in sections at the end. One simple way is to copy this HOWTO in these files as,

```
# cp Swap−Space−HOWTO  /etc/rc.d/init.d/swapinit.sh
# cp Swap−Space−HOWTO  /etc/rc.d/init.d/swaphalt.sh
```

and then using your favourite editor, delete all execpt the required part. Now make these file "executable" by giving the following commands

```
# chmod +x swapinit.sh
# chmod +x swaphalt.sh
```

- Run fdisk and find the size of the swap partition in blocks. In both the above files (swap????.sh) replace the YYYYY by this number.

- Check the name of the shutdown file. For SysVinit this is the file listed in the following line of /etc/inittab; add it if you don't have it.

  ```
  # Runlevel 0 means shut down the system
  l0:0:wait:/etc/brc
  ```

  (Under Red Hat Linux, this file probably will be /etc/rc.d/init.d/halt) If you are not in a position to find it out, browse through the /etc directory and find the initialization file that contains the "swapon −a" command. Only this much for now !

  For the remainder of this file, I will assume the filename was /etc/rc.d/init.d/halt (halt for short).

- Check the name of your system initialization file. For SysVinit, this is the file listed in the following line of /etc/inittab.

  ```
  # System initialization
  si::sysinit:/etc/rc.d/rc.sysinit
  ```

  In case of your distribution, your file may be different. I assume that the file is /etc/rc.d/rc.sysinit (rc.sysinit for short)

- Add the following piece of code to your initialization file (rc.sysinit), in place of the "swapon −a" command. You may consider not deleting but just commenting the original commands so that if something goes wrong, we can restore the script to it's initial state.

  If your swapon is in /etc, replace /sbin/swapon with /etc/swapon. If it is in /bin, replace with /bin/swapon. Do the same for mkswap.

```
# -------------------------------------------------------------------------
# removed by yourname
# Start up swapping.
# action "Activating swap partitions" swapon -a
# -------------------------------------------------------------------------

# ------------------------------------------------------------------------
# added by yourname
# Verify and initialize swap space
#
/etc/rc.d/init.d/swapinit.sh
# ------------------------------------------------------------------------
```

- A simple way to paste the text is to start two Virtual Consoles. Log as a root in both cases. Use your favourite editor to open this HOWTO on one console and the script to be modified on other console. Now select the above code with your mouse and switch to the other console. Place the cursor at the desired position to paste and press the right mouse button. The selected text will be copied.

- Add the following piece of code to your shutdown file (halt).Put this after any command that might need swap to be in place. Normally, your halt file will have the "swapoff −a" command. First replace the −a by /dev/winswap. Then immediately after that section, add the following commands.

  *Note for RedHat Linux 6.1 :* It seems that Red Hat Linux 6.1 halt script automatically finds all the swap partitions and turns of swapping. So, RedHat Linux 6.1 users need **not** replace the `swapoff $SWAPS` comamnd by the `swapoff -a` command. In that case, just add the `/etc/rc.d/init.d/swaphalt.sh` command to the halt file

```
# --------------------------------------------------------------------------
# Modified by yourname
# replaced "swapoff -a" by "swapoff /dev/winswap"
#
# Turn off swap, then unmount file systems.
runcmd "Turning off swap and accounting" swapoff /dev/winswap
[ -x /sbin/accton ] && /sbin/accton
# --------------------------------------------------------------------------

# --------------------------------------------------------------------------
# Added by yourname
# check swap signature and restore Windows swap info
/etc/rc.d/init.d/swaphalt.sh
# --------------------------------------------------------------------------
```

## 6.5 Reenable swapping

Uncomment the line in /etc/fstab that you commented earlier. (Not really necessary, since we now do not refer to fstab for swap partitions). Reboot Linux. You should now have swapping on the new swap device.

---

# 7. A couple of notes

- 

There is no need to add /dev/winswap to your /etc/fstab file. In fact, it is probably wise not to do so (except possibly as a comment).

- If your Linux session crashes or otherwise exits without running /etc/rc.d/init.d/halt, you will need to reboot and exit Linux before swapping in Windows will work. It is also possible to FORMAT X: and re−create the Windows swap file. The only way around this would be to put the equivalent of the /.../halt commands in the DOS AUTOEXEC.BAT file. I have tried to write some programs to do this, but that turned out to be a hell lot of trouble ! First and foremost, simply restoring the total special sectors doesn't please DOS. Along with that, it seems you need to modify some internal data structures so that DOS starts recognizing your drive, otherwise DOS comes out with the error message:

  ```
  Invalid media type reading drive X:
  Abort, Retry or Fail ?
  ```

  Since I am not aware of many details of DOS's or Windows's handling of disk drives, I think at this point, you may consult some DOS expert. Of course, if you are not concerned about the booting time, then you can simply add the command to perform an unconditional FORMAT of X: and then restore the TSS's. May be there are some other solutions. If you are aware of one, please e−mail me today !

- One way to restore your Windows swap partition is to start Linux, disable swapping as described above, and then use the following command :

  ```
  # swapoff −a
  # zcat /etc/winswap.gz > /dev/winswap
  # reboot
  ```

- **Why to delete the RECYCLED folder ?** According to the procedure mentioned above, you must delete the `RECYCLED` folder. This is necessary because we will be saving and restoring only the sectors unto the root directory, but no further sectors. Also, we want the drive X: to be restored to a "clean" state with no errors. If we don't delete the `RECYCLED` folder, then the root directory entry is restored, but the "." and ".." entries for this folder are not restored and ScanDisk finds errors on the disk. To prevent this from happening, we will delete the `RECYCLED` folder and hence only the entry for win386.swp file will be restored.

- If DOS' FDISK reports the partition as a "logical DOS drive", it has a number of 5 (as in /dev/hda5) or higher. It is *NOT* the partition labelled "extended" which has a number of 4 or less!! If your Linux fdisk does not display logical partitions, you have a broken Linux fdisk (Slackware 2.2 included a broken fdisk, for example.) You can try "cfdisk" if your distribution has it, or you will have to get a working fdisk.

- When you turn off swapping, Linux has to manage everything in the main memory itself. If your system memory is low, say something like 16MB, then some large applications that run during initialization may not be able to run properly. As an example, you may get the following message during initialization:

```
Finding module dependencies Out of memory
```

I don't think this will cause any problems later until you reenable swapping. But it's always better to take some precautions. As you will be seeing in the next section, Linux can also be configured to use a swap file instead of a swap partition. So if you feel that Linux will not run properly with your normal system memory, create a swap file that Linux will use temporarily until swapping is restored. This swap file should preferably in the root directory. Figure out the minimum size of swap space required. It will probably equal to your system memory. Create a swap file of that size by giving the following commands (assuming 16MB swap space):

```
# dd if=/dev/zero of=/swap bs=1024 count=16384
# mkswap /swap 16384
# sync
```

Now place the command `swapon /swap` just after the original `swapon −a` command in the rc.sysinit file. Thus, now your system has been configured to use swap file and hopefully there will be no Out of memory errors. After having a workable swap partition (i.e after following through this HOWTO completely), you can remove the `swapon /swap` line that you added to your rc.sysinit file.

- Please do not mail me unless you have checked and re−checked that you copied your scripts correctly. More than half of the problems reported with this Mini−HOWTO have been due to typos when copying the scripts. (Use cut−and− paste if you can!). Even a spacing error can cause an error. For example, if at the startup you get the messages

  ```
  Verifying swap space...Linux signature found
  swapon: /dev/winswap invalid argument
  ```

  Then, surely there has been a typing mistake in your scripts. Please check your scripts.

- Sometimes, the swap partition is detected to have neither Linux nor Windows info and the "No signature found" error appears. I exactly don't know why this happens, but for that case, I have written the code in swap????.sh files to try six times before reporting an error. Normally, the DOS signature should be found on the first try, but many a times on my system it is found on iteration 2, 5 or not at all after 12 iterations. You may consider increasing the number of tries to some value like 12 or 24, there is no harm in doing so. Once on my system, the script could not detect the signatures even after 50 iterations ! If such a situation arises, you may consider making a symbolic link to the swapinit.sh file in your root home directory, and execute it at startup.

  ```
  # ln −s /etc/rc.d/init.d/swapinit.sh swapinit.sh
  # ./swapinit.sh
  ```

  However, what I have observed is that once the setup runs on your system for some time, the appropriate signatures are always found on the first iteration.

- **For added reliability**

Your Linux Box runs something very important, or your system memory is less so that without swapping, your applications just start collapsing ? You have set up the swap partition to be shared by Windows. But many a times, your system is unable to detect the proper signatures after exhaustive probing and so disables any swapping partitions. **At your own risk**, you may consider removing the signature checking logic, but here is a more approachable solution.

First refer to the previous point and create a swap file in the Linux root directory, with the name /swap. Keep it's size according to your requirement. Now open the swapinit.sh file and modify the last `if` statement as follows:

```
if [ $activated −ne 1 ] ; then
  echo "Swap signature not found after $loopcount tries"
  echo "Activating swap file"
  swapon /swap
  exit 1
fi
```

Thus if we find that swap partition lacks appropriate swap signatures, then we enable the swap file. Since the swap file is on the root partition and the Linux Kernel refuses to proceed until it mounts the root file system, you are safely assured that the swap file will always be successfully activated. Instead of creating and keeping the swap file and wasting disk space, you may also modify the code to create the swap file *on the fly*.

- **Saving more disk space** As we saw, Linux and Windows can share their swap spaces. But Window doesn't really require as large a swap file as your Linux swap partition ? Then you can have some free space on the swap partition (keep the Windows swap file size smaller than the swap partition size). Swapping for Windows based applications is managed by Windows itself. But many DOS programs like Turbo C++ Compiler, Norton DOS etc. also manage their own swap files. In that case it is possible to configure these programs so that they create their swap files in the swap partition instead of their default location. This will increase your usable disk space. Also, most of these program forget to delete the swap files they created, leading to a large used disk space. But with this setup, every time you shut down Linux, you get a clean swap partition, saving your time in finding and deleting these swap files. For example, you can tell Turbo C++ to swap on drive say E: by invoking it as

  ```
  C:\CLAN>TC /RE
  ```

  For Norton DOS, set the variable in the NDOS.INI file as,

  ```
  Swapping = E:\
  ```

  Many other programs may rely on the `TEMP` or `TMP` environment variables to decide where they will swap. In that case you may set these variables appropriately in your AUTOEXEC.BAT file.

- **What if the swap partition is a FAT32 ?** No, you can't have a swap partition with a FAT32 file system. First of all, Linux can use a maximum of 128MB of swap space on a single swap partition. Thus, if your swap partition is larger than 128MB, you are wasting your disk space. And Windows will refuse to convert a partition of 128MB into a FAT32 file system.

-

**The new style swap space** Linux knows about two styles of swap areas, the old style and the new style. By default, you will be using the old style swap space with a signature "SWAP−SPACE". If you want to use the new style swap space, then first of all change "SWAP−SPACE" in all the above scripts to "SWAPSPACE2". Also add an option −v1 in the mkswap command in the swapinit.sh script, as follows

```
/sbin/mkswap −v1 /dev/winswap YYYYY
```

Also, run the following commands from the shell prompt :

```
# swapoff /dev/winswap
# mkswap −v1 /dev/winswap YYYYY
# swapon /dev/winswap
```

It seems that under new style swap space, your usable swap partition can be as large as 2GB on i386 platforms. This means that you can now have a swap partition larger than 512MB which Windows can format as a FAT32 partition . Someone help on how do we save the "Total Special Sectors" under a FAT32 partition.

- **For users of WINE** WINE is a program which allows running Microsoft Windows programs on Linux. If you are using WINE on a system that has been set up to share swap partition, then when running a Windows program a `win386.swp` file of about 8kb is created in your Windows folder. This is because the swap file on X: is not available at that time. You can safely delete this file once you finish running your Windows programs.

---

# 8. What are we exactly doing ?

- Both Windows and Linux use swap space. When an OS is shut down, the contents of the swap space are just garbage, the OS doesn't bother about what is in it. Now what is the point in devoting a 60MB partition for swap space to Linux and about 32MB of disk space separately to Windows when only one of them is going to run at a time (It is possible to run both Linux and Windows at the same time, thanks to VMWare. For more information, see VMWare website). So we want Linux and Windows to share the same hard disk space as swap space.

- The problem is that Linux uses a partition as a swap space and Windows uses a file as a swap space. ( Although it is possible to create a swap file for Linux, it is not recommended. See *"Linux Installation and Getting Started"* by Matt Welsh). Now if Linux uses the same partition on which Windows stores it's swap file, it will overwrite the boot sector, FAT and other data tables that Windows assumes to exist on every drive. Thus, Windows will not be in a position to find the swap file on startup and will create the win386.swp file in the windows directory. Thus, even if by chance, your Linux crashes, Windows will be in a position to start. In such a case ( with Windows swap file in the Windows directory), just restore the DOS/Windows swap info by restarting Linux and the start Windows. Now Windows will be using the swap file on drive X: so you can safely delete the win386.swp file in the windows directory.

-

The remedy is to store those critical data tables once on the Linux partition in some file. Then every time Linux starts, check whether the swap space was last used by Linux or Windows (using the label you gave to your partition). If it was Linux, just enable swapping else first make a swap partition on the X: drive using "mkswap /dev/winswap" and then enable swapping.

- When you shut down Linux, it is important to restore the swap partition to the DOS format, so that Windows can start properly. For that purpose we add the

  ```
  /bin/zcat /etc/winswap.gz > /dev/winswap
  ```

  command to the halt file.

---

# 9. The swapinit.sh script

This shell script initializes the swap space on the partition. The code for signature detection has been repeated 6 times because many a times the signature is properly detected only on the second or the third try.

(Adapted from the original Swap−Space−HOWTO by H. Peter Anvin)

```
#!/bin/sh
#
# swapinit.sh - activate the swap partition
#
# written by Rahul U. Joshi
# Verify and initialize swap space
#

echo −n 'Verifying swap space... '

loopcount=0

# flag to indicate whether the partition has been activated or not
activated=0

# check for signatures 6 times before giving up
while [ $loopcount −lt 6 ]
  do

  if [ "`/bin/dd 2>/dev/null if=/dev/winswap bs=1 count=10  skip=4086`" = 'SWAP−SPACE' ]; ther
    echo "Linux signature found, iteration $loopcount"
    echo "Activating swap partitions"
    swapon /dev/winswap
    activated=1
    break

  elif [ "`/bin/dd 2>/dev/null if=/dev/winswap bs=1 count=11 skip=43`" = 'SWAP SPACE ' ]; ther
    echo "DOS signature found, iteration $loopcount"
    echo "Making swap partition"
    mkswap /dev/winswap YYYYY
    echo "Activating swap partitions"
    swapon /dev/winswap
    activated=1
    break
```

```
  else
   let loopcount=loopcount+1
  fi

  done


if [ $activated −ne 1 ] ; then
  echo "Swap signature not found after $loopcount tries"
  echo "No swapping partitions activated"
  exit 1
fi
```

# 10. The swaphalt.sh script

This script first checks the Linux swap signature and then restores the Windows file system on it.

(Adapted from the original Swap−Space−HOWTO by H. Peter Anvin)

```
#!/bin/sh
#
# swaphalt.sh   This file is executed through the /etc/rc.d/init.d/halt
#               script after swapping and accounting has been turned off.
#
# Author:       Rahul U. Joshi
#

# check swap partition signature and restore Windows swap info

loopcount=0

# flag to indicate whether the swap info has been restored or not
restored=0

# check for swap signature 3 times before giving up
while [ $loopcount −lt 3 ]
  do

  if [ "`/bin/dd 2>/dev/null if=/dev/winswap bs=1 count=10 skip=4086`" = 'SWAP−SPACE' ]; then
    echo "Restoring DOS/Windows swap info , iteration $loopcount"
    /bin/zcat /etc/winswap.gz  >  /dev/winswap
    restored=1
    break
  else
    loopcount=loopcount+1
  fi

  done

if [ $restored −ne 1 ] ; then
   echo "Swap signature not found after $loopcount tries"
   echo "Skipping restoring"
fi
```

# 11. The msinfo.sh script

This shell script analyses the boot sector of the given partition and displays some information along with the "Total Special Sectors" in a message box. It assumes that the filesystem on the given partition is a FAT16. If not, it will print an error message and exit. Invoke it as

```
# msinfo <partition name>
```

To run this script, you will need the "dialog" program that displays dialog boxes. You can get it from here.

```
#!/bin/sh
#
# msinfo.sh           This shell script displays the boot sector of the
#                     given partition.
#
# Author:             Rahul U. Joshi
#
# Modifications       Removed the use of expr and replaced it by the let
#                     command.
#
# -------------------------------------------------------------------------
# This program is a free software, you can redistribute it and/or modify
# it under the eterms of the GNU General Public Liscence as published by
# the Free Software Foundation; either version 2 or (at your option) any
# later version.
#
# This program is being distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY without even the implied warranty of
# MERCHANTIBILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General
# Public Liscence for more details.
# --------------------------------------------------------------------------


# check for command line arguments
if [ $# -ne 1 ]; then
   echo "Usage: msinfo <partition name>"
   exit 1
fi

# check whether the input name is a block device
if [ ! -b $1 ]; then
   echo "msinfo: $1 is not a block device"
   exit 1
fi

# create two temporary files for use
TMPFILE=`mktemp -q /tmp/$0.XXXXXX`
if [ $? -ne 0 ]; then
    echo "msinfo: Can't create temp file, exiting..."
    exit 1
fi

TXTFILE=`mktemp -q /tmp/$0.XXXXXX`
if [ $? -ne 0 ]; then
   echo "msinfo: Can't create temp file, exiting..."
   rm -f $TMPFILE
   exit 1
```

```
fi

backtitle="`printf "%78s" "msinfo, Information about FAT16 filesystem -- Rahul Joshi"`"

dialog --title "Boot sector of $1" --backtitle "$back_title" \
       --infobox "\nAnalysing boot sector for $1\nPlease wait..."  14 60

# truncate TXTFILE to zero length
echo > $TXTFILE

# get Formatting DOS version
dd 2>/dev/null if=$1  bs=1 count=8 skip=3 | dd 2>/dev/null of=$TMPFILE
printf >>$TXTFILE "%30s : %s\n" "Formatting DOS version" "`cat $TMPFILE`"


# get file system
dd 2>/dev/null if=$1  bs=1 count=8 skip=54 | dd 2>/dev/null of=$TMPFILE
printf >>$TXTFILE "%30s : %s\n" "Filesystem" "`cat $TMPFILE`"

# check if filesystem in a FAT16
if [ "`cat $TMPFILE`" != "FAT16   " ]; then
  dialog --title "Boot sector of $1" --backtitle "$back_title" \
         --infobox  "\nCan't find a FAT16 filesystem on $1"  14 60
  exit 2
fi

# get volume label in boot sector
dd 2>/dev/null if=$1  bs=1 count=11 skip=43 | dd 2>/dev/null of=$TMPFILE
printf >>$TXTFILE "%30s : %s\n" "Volume label in boot sector" "`cat $TMPFILE`"


# get Sector size
dd 2>/dev/null if=$1  bs=1 count=2 skip=11| od -An -tdS | dd 2>/dev/null of=$TMPFILE
printf >>$TXTFILE "%30s : %d\n" "Sector size" `cat $TMPFILE`
sector_size=`cat $TMPFILE`


# get Reserved sectors
dd 2>/dev/null if=$1  bs=1 count=2 skip=14| od -An -tdS | dd 2>/dev/null of=$TMPFILE
printf >>$TXTFILE "%30s : %d\n" " Reserved sectors" `cat $TMPFILE`
reserved_sectors=`cat $TMPFILE`


# get FAT sectors
dd 2>/dev/null if=$1  bs=1 count=1 skip=16| od -An -tdS | dd 2>/dev/null of=$TMPFILE
fat_count=`cat $TMPFILE`

dd 2>/dev/null if=$1  bs=1 count=2 skip=22| od -An -tdS | dd 2>/dev/null of=$TMPFILE
sectors_per_fat=`cat $TMPFILE`

# calculate the no of sectors allocated for FAT's
let fat_sectors=fat_count*sectors_per_fat

printf >>$TXTFILE "%30s : %u (%u x %u) \n" "FAT sectors" "$fat_sectors" \
        "$fat_count" "$sectors_per_fat"


# get root directory sectors
dd 2>/dev/null if=$1  bs=1 count=2 skip=17| od -An -tdS | dd 2>/dev/null of=$TMPFILE
root_sectors=`cat $TMPFILE`

# calculate the no of sectors allocated for root directory
```

```
let root_sectors=root_sectors*32/sector_size

printf >>$TXTFILE "%30s : %u\n" "Root directory sectors" "$root_sectors"


# get Total special sectors
let total=reserved_sectors+fat_sectors+root_sectors
printf >>$TXTFILE "%30s : %u\n" "Total special sectors" "$total"


# display the information
dialog --title "Boot sector of $1"  --backtitle "$back_title"  --msgbox "`cat $TXTFILE`" 14 60

# delete temporary files
rm -f $TMPFILE
rm -f $TXTFILE

# end of msinfo.sh
```

---

## 12. The original msinfo file

Here is the msinfo file required to find the "Total Special Sectors". It was included by the original author of this HOWTO in an encoded form. First copy the text from the line "begin 755 msinfo.gz" to the line "end" into a temporary file say temp.uu. Then uudecode this file, and finally gunzip to get msinfo file.

```
# uudecode temp.uu
# gunzip  msinfo.gz
```

Now run the program using the command

```
# ./msinfo /dev/winswap
```

The program displays the boot sector information for given partition as well as the "Total Special Sectors" for the drive.


```
begin 755 msinfo.gz
M'XL(`$$YNRT"`Y557TQ;51@_MW]HZ0AM'"$U-.28W([..>84%1%D2QH>',9*
M2S&VC);V=FV]T*;WWBD+1I)K'YH;DOJF#YH80GPR^J""D"R8-K('MS2;#Q*S
M%Y,E-P&592A$V:[?.;?M"FP/?DG/G^_[?7_Z_3G7PL313TT(?8P0<J"CI':8
M$5H^#:=5<KT^MQ`^UAQ1O*QUH0<.]B7S/FC[U!98Y;))]<*N^AG"^LB((J%E
MAFIM[A9U0O`;%:*7N#[L%G!_-IH34V(J,WUZ.CK%O6I#N4DTE,E-144Q-7T)
M>]_PX\M<3@!`'ZZ26[`-I7A.F!%$;JK.;21`!#.\-,5A/CK)\=B3FL:3F8PH
M<#$QDSO91Q%^>L%"Z@KW*"-NR3;""5SN,A?'NI[0=P0Q=#[P&&$5@3UNZ3VW
M=-(V`NYQ/)6CX)DC2F`KD!&C/!:R7"Q%]L-6`8'D+=]8T*^$V8@RR\;5\P9(
M^;\&T:F]E`Q`EI,_6YHCZBT#BJ09M0VV9!:XVKLF]0NX%'Z4R];\W](3A-%J
MI:`_&12!>@33QC0S'V8]?O5+0`;3UC23K)!"?@97N8R#:29MF/>R)PI>UJF)
MK%/-4QR39OSJE8,8MHZ)UC#`?1ZX'N!ZU`L'X;@.?X["C<1DUT%,9QW32C''
M"<98PQR#8#=(L'>96NQWR/4W1@>H3\'!OC3,MMN_"[,=A1OVK^XEH($[$S^H
MY@28=MB78-$/`=8)(L=\P*$5*I_LS@)_G7!=\Z9V@F"``Y$XQM[T!T>(WHNU
M/T9CH)YIZI("&9ZK32AR=DTN/QNZ."&7?27H_<(=_^=Q"U1N3Y.>!-E`$BY(
M;(4ZG)/[\_0F_54HY7\5[<!33)\2M%:Y(.THX_N%K?%2L9&(O=4>8F^@IV9(
M,<HEDU:17T'23M)GA#ANW-<F'J5-]>6!G2;BU+YN)@<RL.JBIFD4[:<###UG
MO3Z77WO?L?O+@@G\+(-<"X&QT"&#_N7@_\2;JGAEF&U1`JR#ZCD/Z]7HJ'ZH
M09_JNAZGJ^N?X5.39_AXMY!!?3@6G>X2,9^)QG%\!IZ@5`SSJ>EWN!SN>HCK
```

```
MLJ%F08R*J5B4YV=T0-R&J#UYJW>4#"7/#D,0/GG`0ZI@__`;6->]K!>C[<GD
M31/IQP>:!B]D?DT,*<;Y6=9Y]3:P][^W`?+>8F(1!*ZY7B1:BN=@D_XY9?0I
M'?ZT0=V@BBWIIV$47#Y@7-,9Q>0<*=7;NMW-NS2>9#_X3V)8()?A$;!`ALRW
M#&\%RO_^@9F6TP<]'M8J@'AFA"[JM]0(WCQ!`[.826!F&A@HG4JWU5R_KKL&
M9GNQF$;J[?NZ\YUZ?B].C,OEWI*\94J`:6LB?U,RK^2.([0QRQK(W!D2[K5M
M,G>&0B7A'K03F"'17*&L#9XU@+U2L9K4-GCIG(K(NE:SM,>SM,?;X,T:&U6,
MT./E%NAUTNE_$,^E(A0M!B7K[D'>X6'\FC25Q3W=+V3YLR^CXJW.S0>#54*#
M#ZE^AI+N-35^_>![U4!?@VQ%ET<L!BC>MC5"\GH->"T6'4/2NS>((N0,DT>_
9K6^!S%65KRS`,`,UVB43\!]-BKG]B`<``#5"
`
end
```


# 13. Acknowledgements, feedback and dedication

This Mini−HOWTO has been largely derived from the Swap−Space Mini HOWTO by **H. Peter Anvin**. I have added the Windows 95/98 specific details as well as a few Red Hat Linux specific details. The msinfo program given in the original HOWTO didn't work on my Red Hat Linux 6 system, so I created a similar program. However, I have also included the original msinfo program. Thanks to all those **great people involved with the development of Linux**,to the people at **Red Hat** for giving a great Linux distribution and to **Sanjiv Guha**, one of the co−authors of "Red Hat Linux 6 Unleashed" from Techmedia for guidance about shell programming. Thanks also to **Matt Welsh**, **Greg Hankins** and **Eric S. Raymond**, the authors of SGML−Tool's Users Guide, which helped me in converting the document in SGML format. Thanks are also to **Peter Norton** and **John Goodman** for there venerable "Peter Norton's Inside the PC", which helped me about DOS file system structure. Finally, thanks to **PC Quest Linux Project** for bringing out Linux CDROMS frequently.

I rely on you, the readers, to make this HOWTO useful. If you have any suggestions, comments, corrections, requests, flames, etc., feel free to contact me. Right now, I have no internet access, but still you contact me in case of questions at my Hotmail account (Hey! Don't blame me for using a Microsoft service. Hotmail was created by our Indian guy Sabeer Bhatia). May be someone from you local place may also come to your rescue.If you find my account unoperational and your problem is indeed important, you may mail me at amitanatu@sprynet.com.

Finally, I would like to dedicate this work to **Robert Lafore**, the author of excellent books "C programming using Turbo C++" and "Object Oriented Programming using Turbo C++".

Enjoy Linux,

Rahul :−)