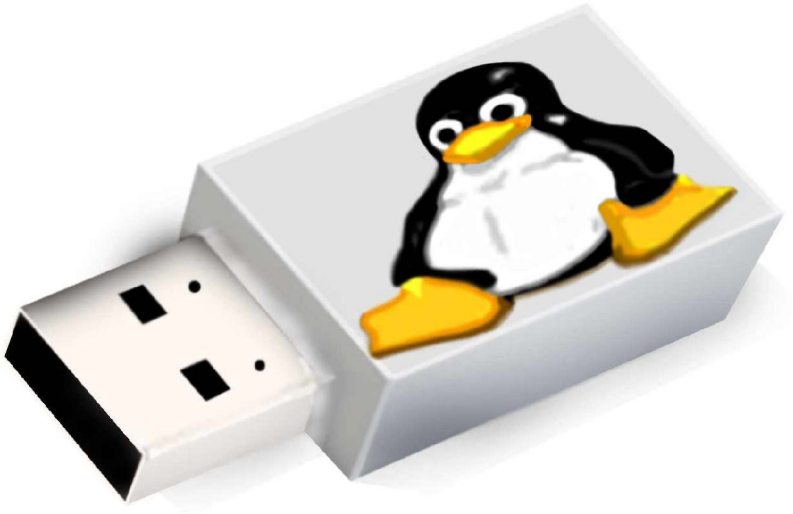
RUNT LINUX



A review of RUNT Linux and other pen drive based linux distributions.

RUNT 4.0 Released Now!

- http://www.ncsu.edu/project/runt
- Based off of Slackware 10.0
- New scripts to make booting easier.
- Kernel 2.4.26
- New logo.
- Keyboard layout selection

What is a Pen Drive Distribution?

- A complete Linux operating system.
- Stored and run entirely from a USB pen drive.
- Capable of booting directly off the pen drive in computers that support it.
- Capable of booting off a floppy disk in computers that do not support USB booting.

Why a pen drive based distribution?

- Portable – Take it anywhere.
- Consistent – Runs the same on every system.
- Writable – Save configuration and files without the use of an additional device.
- Pen drive can be used for other purposes as well.
- Decreasing cost of Flash Memory.

Prices for SanDisk Cruzer at Newegg.com (11/11/2004)

• 128 MB	$24.50	$0.190/MB
• 256 MB	$36.00	$0.141/MB
• 512 MB	$49.00	$0.095/MB
• 1 GB	$79.00	$0.077/MB

Common Uses

- Personal secure linux system you can run anywhere.
- Testing utility

A Few Pen Drive Distributions

- Knoppix Based
  - Feather Linux
  - Damn Small Linux (DSL)
  - Flonix
- Puppy Linux
- Slax
- RUNT

Feather Linux

- UK Based Knoppix remaster.
- CD and USB edition.
- Graphical interface with Fluxbox.
- Fits in 64 MB.
- Can save /home and settings like Knoppix.
- Unzip and run syslinux to install.
- Boot floppies available.

Damn Small Linux

- US Based Knoppix remaster.
- Fits in 50 MB.
- Graphical interface with Fluxbox.
- Requires that you boot from CD and be connected to internet to install to USB.
- Easy to add additional software while running.
- Can save /home and settings like Knoppix.

Flonix

- French based DSL remaster.
- Previously had CD and USB versions.
- Recently went 1.0, now non-free.
- Must be purchased on a pen drive. $37
- Graphical interface with IceWM.

Puppy Linux

- Independent Australian distribution.
- Runs entirely in RAM.
- Requires CD download to install to pen drive.
- Graphical interface with FVWM95.
- Fits in 53 MB
- Can either mount pen drive as /root using umsdos or creates ext2 loop FS.

Slax

- Czech Republic, Slackware based CD.
- Supposedly installable to pen drive.
- Script failed for me.
- Starts in text mode, can load KDE or Fluxbox.
- Requires 180 MB.

RUNT

- Based on Slackware's zipslack.
- Designed around USB only.
- Easy install. Just unzip onto a pen drive.
- Fits in 128 MB.
- Complete text interface.
- UMSDOS filesystem.
- Designed as a testing/recovery tool.

History of RUNT

- **R**esNet **U**SB **N**etwork **T**ester
- Designed for NC State ResNet in 6/'02
- Testing tool for technicians to use to diagnose problems.
- Provided better abstraction of hardware and software.

Advantages of RUNT

- Like a normal Slackware installation
  - Easy to install additional packages
  - Software, settings, files always saved.
- Easy to install
- Dynamic size usage
- Complete text interface. Most programs you'd expect to be able to use are already installed, and if not they are easy to add.

Disadvantages of RUNT

- Inefficient use of space because of UMSDOS.
- Frequently writes to flash. Can wear out a pen drive.
- Lacks some software by default.

UMSDOS

- Msdos filesystem with linux extensions
- Stores extended data in –linux-.--- in each directory.
- Contains entire root filesystem in /linux, so it doesn't have to be the only thing on the drive.
- When viewed as an MSDOS filesystem, filenames are mangled to 8.3 format.

Standard Linux Booting

- Boot loader loads kernel into memory and executes.
- Kernel mounts root filesystem and runs init.

INITial RamDisk (initrd)

- Needed when the kernel is unable to mount the root filesystem.
- Sometimes easier to create a generic kernel and use an initrd to load appropriate modules for specific hardware and filesystems.
- Can be used to perform functions the kernel cannot such as scripting prior to mounting /

Initrd structure

- Usually a small gzipped ext2 filesystem.
- Contains modules, programs and scripts that need to be run before mounting root.

Building an initrd

```
dd if=/dev/zero of=initrd bs=400k count=1
mke2fs -F -m0 initrd
mount -t ext2 -o loop initrd foo/
```

- Copy in files, and devices as needed.

```
umount initrd
gzip -9 initrd
```

Boot process with initrd

1) Boot loader loads kernel and initrd into memory.
2) Kernel converts initrd into a RAM disk and frees memory used by initrd.
3) Initrd mounted read-write as root
4) /linuxrc is executed uid 0
5) One of two methods is used to mount the new root filesystem.

change_root method

1)/linuxrc exits

2)Kernel mounts real root device set by rdev in the kernel or by root= on the boot command line.

- This method is considered depracated, but nonetheless it is simple and works.

pivot_root method

1)/linuxrc mounts real root filesystem

2)/linuxrc places the root filesystem at / using the system call pivot_root.

3)Kernel runs init from new root filesystem

4)Initrd filesystem is removed

- More complex, but preferred method.

Requirements to mount USB as root filesystem

- BIOS support for USB Booting, or another method of loading the kernel and initrd such as a boot floppy.
- A kernel with support for USB storage. This can be handled with an initrd.
- A solution to the following problem...

Problem booting USB

- While kernel supports USB storage devices, it doesn't have time to initialize them before attempting to mount them as root.

Solutions

- Hack the kernel
  - http://www.lammerts.org/software/kernelpatches/
  - I haven't found these hacks to consistently work
- Use an initrd to force the kernel to wait before mounting the device as /

My initrd

- Busybox compiled with just a few functions
  - lash shell
  - echo
  - insmod
  - sleep
- Compiled with uClibc to keep size down. (76K)

My initrd (cont'd)

- Modules needed to load usb storage device
  - Usbcore
  - Host controller modules
    - Uhci
    - Usb-ohci
    - Ehci-hcd
  - Usb-storage
- Linuxrc script to load modules...

My linuxrc

```
#!/bin/lash
echo "Loading usbcore module"
/sbin/insmod /lib/usbcore.o
echo "Loading uhci module"
/sbin/insmod /lib/uhci.o
echo "Loading usb-ohci module"
/sbin/insmod /lib/usb-ohci.o
echo "Loading ehci-hcd module"
/sbin/insmod /lib/ehci-hcd.o
echo "Loading usb-storage module"
/sbin/insmod /lib/usb-storage.o
echo "Sleeping 5 so USB driver can initialize a
drive"
sleep 5
```

Getting the kernel and initrd into memory. Bootloaders

- LILO,GRUB
  - These run from the MBR or superblock of a linux filesystem, loading the files from within the linux filesystem.
- Syslinux
  - Runs from the first sector of an msdos filesystem, loading the files from within the filesystem.
- Loadlin
  - MSDOS program to load files.

RUNT's bootloaders

- RUNT can use syslinux, loadlin, or LILO
- Syslinux is preferred, and can be run from a boot floppy or the pen drive itself. Scripts are included.
- Loadlin can be used if your pen drive is DOS bootable
- LILO can also be installed on the drive.

Scripts to make bootable

- New with RUNT 4.0
- Scripts are in /runtutil on pen drive
- makeboot.{sh,bat}
  - Makes pen drive bootable
- mkfloppy.{sh,bat}
  - Creates boot floppy

makeboot

- Copies necessary syslinux files to root of pen drive
- Installs syslinux onto drive
- Installs an appropriate MBR to the pen drive.

mkfloppy

- Formats a floppy disk
- Installs syslinux on the disk
- Copies files to disk

Links

- Home page
  - http://www.ncsu.edu/project/runt
- Forum
  - http://moses.sca.ncsu.edu/phpBB2/index.php
- Bugzilla
  - http://moses.sca.ncsu.edu/bugzilla/
- Contributions
  - http://home.earthlink.net/~joelebel/
- Freshmeat Project Details
  - http://freshmeat.net/projects/runt/

Questions?