Mastering Gentoo

Add the meaningful to the Wiki

Onkobu Tanaake

2020-11-22T11:00:00

The common desktop Linux variants like Fedora, Ubuntu or Debian have an installer that gets the average computer up and running within a few hours. Its result is an absolutely usable desktop environment. But Linux is so much more. When using Linux From Scratch, Arch Linux or Gentoo you end up touching a mirror turning your finger into liquid metal. Either panick now or see behind the Matrix. This is the first part of 4 starting with suggestions for filesystem layout.

You can run any Linux fine without any swap. I normally provide 4GByte of swap if it is a desktop system. When hibernating an image of the system RAM is put there. Normally I supply 16GByte of RAM. Servers get a lot more, depending on the purpose. All my Raspberry PIs run without any swap unless they have a harddisk attached. None of them is a desktop system but a dedicated and often headless worker. Their configuration is tweaked so that for example a web server's cache fits into the RAM. So no swap allows the SD card to be read only – requiring minor tweaks of logging since system state under /run is already a tmpfs. Then it is safe to simply turn off the Raspberry PI without any corruption of the SD card. All USB disks have a journaling file system.

According to Linux Standard Base I always move distfiles and binary packages to /var which is always a separate partition of 40GByte. Monitoring of such a separate partition is easy and as soon as it reaches 70% usage a job wipes it. This saves the entire system from blocking due to lack of disk space to create temporary files or such.

All my portable computers have no extra partition for /usr. With a single built-in harddisk it doesn't make any sense to split root or / and /usr. But they have a separate /var partition. All full size PCs have split partitions and follow the suggestions of Early Userspace Mount [https://wiki.gentoo.org/wiki/Early_Userspace_Mounting]. This saved me from at least two failures since I can mount root itself read-only and repair/ restore journal of /usr seperatly. Also /boot is never mounted during regular operation. I use the following sizes for the partitions:

- / 10GByte
- /boot 256MByte – easily holds 4 kernel variants and EFI directory
- /usr 10-40GByte, on Raspberry PIs even 4GByte with more cleaning
- /var like /usr
- /srv different mount points for servers, e.g. /srv/postgres, /srv/svn or /srv/http, often RAIDs

- /tmp 1GByte tmpfs, Gentoo default
- /home always a separate disk with encryption or the remaining space (portable computers)
- swap – only if necessary/ wanted, never larger than 4GByte, always tmpfs/ RAM disk
- /run – automatically a tmpfs on Gentoo

These days all remaining mount points holding system state like /tmp or /run come as tmpfs. So they can't pile up beyond reboots and can also be monitored quite easily. I also use a tmpfs for /var/tmp/portage enabling builds to happen in RAM. Nowadays this requires 10GByte of space for builds of Rust. Libreoffice and web browsers get along with 8GByte.

Upcoming parts:

- UEFI Boot, chroot, /boot and grub, mount-script to the rescue
- Solving common build errors
- Chroot Qemu to build Raspberry PI packages on powerful hardware

See also:

- Gentoo [https://onkobutanaake.de/it/gentoo_expert.html]
- Linux is an Alternative [https://onkobutanaake.de/it/linux.html]