Conception OS - Commands

Preparing the disks

Partitioning the disk with GPT for UEFI

Creating a new disklabel / removing all partitions

```
fdisk /dev/sda

g
```

Creating the EFI system partition (ESP)

```
n
1
[enter]
+256M

t
1
```

Creating the swap partition

```
n
2
[enter]
+4G

t
2
19
```

Creating the root partition

```
n
3
[enter]
[enter]
```

Saving the partition layout

```
w
```

Creating file systems

Applying a filesystem to a partition

```
mkfs.vfat -F 32 /dev/sda1
mkfs.ext4 /dev/sda3
mkfs.ext4 /dev/sda2
```

Mounting the root partition

```
mount /dev/sda3 /mnt/gentoo
```

Installing the Gentoo installation files

Installing a stage tarball

Downloading the stage tarball

```
cd /mnt/gentoo
// gérer les mises à jour pour récupérer la dernière
wget https://mirror.init7.net/gentoo/releases/amd64/autobuilds/current-stage3-amd64-
```

Unpacking the stage tarball

```
tar xpvf stage3-*.tar.xz --xattrs-include='*.*' --numeric-owner
```

Configuring compile options

```
nano -w /mnt/gentoo/etc/portage/make.conf
// rajouter MAKEOPTS pour parralléliser le tout
// MAKEOPTS="-j2"
// CTRL S + CTRL X
```

Installing the Gentoo base system

Chrooting

Copy DNS info

```
cp --dereference /etc/resolv.conf /mnt/gentoo/etc/
```

Mounting the necessary filesystems

```
mount --types proc /proc /mnt/gentoo/proc
mount --rbind /sys /mnt/gentoo/sys
mount --make-rslave /mnt/gentoo/sys
mount --rbind /dev /mnt/gentoo/dev
mount --make-rslave /mnt/gentoo/dev
mount --bind /run /mnt/gentoo/run
mount --make-slave /mnt/gentoo/run

test -L /dev/shm && rm /dev/shm && mkdir /dev/shm
mount --types tmpfs --options nosuid,nodev,noexec shm /dev/shm
chmod 1777 /dev/shm
```

Entering the new environment

```
chroot /mnt/gentoo /bin/bash
source /etc/profile
export PS1="(chroot) ${PS1}"
```

Mounting the boot partition

```
mount /dev/sda1 /boot
```

Configuring Portage

Installing a Gentoo ebuild repository snapshot from the web

```
emerge-webrsync
```

Choosing the right profile

```
eselect profile list

eselect profile set <NUMBER | PROFILE_NAME>
```

Updating the @world set

```
emerge --ask --verbose --update --deep --newuse @world
```

Configuring the USE variable

```
// rien a faire, juste output pour vérifier
```

Configuring the ACCEPT_LICENSE variable

```
nano /etc/portage/make.conf
// mettre: ACCEPT_LICENSE="*"
```

Configure locales

Locale generation

```
nano -w /etc/locale.gen

en_US ISO-8859-1
en_US.UTF-8 UTF-8
de_DE ISO-8859-1
de_DE.UTF-8 UTF-8

locale-gen
```

Locale selection

```
nano /etc/env.d/02locale

LANG="de_DE.UTF-8"
LC_COLLATE="C.UTF-8"

env-update && source /etc/profile && export PS1="(chroot) ${PS1}"
```

Configuring the Linux kernel

Kernel configuration and compilation

Installing the sources

```
emerge --ask sys-kernel/gentoo-sources

eselect kernel list
eselect kernel set <NUMBER>
```

Manual configuration

Introduction

```
emerge --ask sys-apps/pciutils

cd /usr/src/linux
make menuconfig
```

Activating required options

KERNEL **Enabling Gentoo-specific options**

```
Gentoo Linux --->
  Generic Driver Options --->
    [*] Gentoo Linux support
    [*]   Linux dynamic and persistent device naming (userspace devfs) support
    [*]   Select options required by Portage features
        Support for init systems, system and service managers  --->
          [*] OpenRC, runit and other script based systems and managers
          [*] systemd
```

KERNEL **Enabling devtmpfs support**

```
Device Drivers --->
  Generic Driver Options --->
    [*] Maintain a devtmpfs filesystem to mount at /dev
    [*]   Automount devtmpfs at /dev, after the kernel mounted the rootfs
```

KERNEL **Enabling SCSI disk support**

```
Device Drivers --->
   SCSI device support  --->
      <*> SCSI disk support
```

KERNEL **Selecting necessary file systems**

```
File systems --->
  <*> Second extended fs support
  <*> The Extended 3 (ext3) filesystem
  <*> The Extended 4 (ext4) filesystem
  <*> Reiserfs support
  <*> JFS filesystem support
  <*> XFS filesystem support
  <*> Btrfs filesystem support
  DOS/FAT/NT Filesystems  --->
    <*> MSDOS fs support
    <*> VFAT (Windows-95) fs support

  Pseudo Filesystems --->
    [*] /proc file system support
    [*] Tmpfs virtual memory file system support (former shm fs)
```

KERNEL **Selecting PPPoE necessary drivers**

```
Device Drivers --->
  Network device support --->
    <*> PPP (point-to-point protocol) support
    <*>   PPP support for async serial ports
    <*>   PPP support for sync tty ports
```

KERNEL **Activating SMP support**

```
Processor type and features  --->
  [*] Symmetric multi-processing support
```

KERNEL **Activating USB support for input devices**

```
Device Drivers --->
  HID support  --->
    -*- HID bus support
    <*>   Generic HID driver
    [*]   Battery level reporting for HID devices
      USB HID support  --->
        <*> USB HID transport layer
  [*] USB support  --->
    <*>     xHCI HCD (USB 3.0) support
    <*>     EHCI HCD (USB 2.0) support
    <*>     OHCI HCD (USB 1.1) support
```

KERNEL **Selecting processor types and features**

```
Processor type and features  --->
   [ ] Machine Check / overheating reporting
   [ ]   Intel MCE Features
   [ ]   AMD MCE Features
   Processor family (AMD-Opteron/Athlon64)  --->
      ( ) Opteron/Athlon64/Hammer/K8
      ( ) Intel P4 / older Netburst based Xeon
      ( ) Core 2/newer Xeon
      ( ) Intel Atom
      ( ) Generic-x86-64
Binary Emulations --->
   [*] IA32 Emulation
```

KERNEL **Enable support for GPT**

```
-*- Enable the block layer --->
   Partition Types --->
      [*] Advanced partition selection
      [*] EFI GUID Partition support
```

KERNEL **Enable support for UEFI**

```
Processor type and features  --->
    [*] EFI runtime service support
    [*]   EFI stub support
    [*]     EFI mixed-mode support

Device Drivers
    Firmware Drivers  --->
        EFI (Extensible Firmware Interface) Support  --->
            <*> EFI Variable Support via sysfs
```

Compiling and installing

```
make && make modules_install
make install
```

Kernel modules

Configuring the modules

```
find /lib/modules/<kernel version>/ -type f -iname '*.o' -or -iname '*.ko' | less

mkdir -p /etc/modules-load.d
nano -w /etc/modules-load.d/network.conf
// put all the names of the modules listed in the file
```

Configuring the system

Filesystem information

Creating the fstab file

```
nano -w /etc/fstab
```

FILE `/etc/fstab` **A full /etc/fstab example**

```
/dev/sda1   /boot        ext4    defaults,noatime     0 2
/dev/sda2   none         swap    sw                   0 0
/dev/sda3   /            ext4    noatime              0 1

/dev/cdrom  /mnt/cdrom   auto    noauto,user          0 0
```

Networking information

Host and domain information

OpenRC

```
nano -w /etc/conf.d/hostname
// change your hostname
// hostname="<NAME>"

// if necessary change domain name
nano -w /etc/conf.d/net
// dns_domain_lo="<NAME>"

// fi necessary set NIS domain name
nano -w /etc/conf.d/net
// nis_domain_lo="<NAME>"
```

systemd

```
hostnamectl hostname <NAME>
```

Problème: je n'ai pas accès à `hostnamectl` , ni `sudo` pour l'installer

Network

DHCP via dhcpcd (any init system)

```
emerge --ask net-misc/dhcpcd

// problème: wget unable to resolve host address
// solution: nano /etc/resolv.conf
//                       y mettre 'nameserver 8.8.8.8' en 1e ligne

rc-update add dhcpcd default
rc-service dhcpcd start

// problème: start-stop-daemon is already running
// solution:

systemctl enable --now dhcppcd

// problème: systemctl commmand not found (et toujours pas accès à sudo)
// solution:
```

netifrc (OpenRC)

```
emerge --ask --noreplace net-misc/netifrc
nano -w /etc/conf.d/net
// put config_eth0="dhcp"

cd /etc/init.d
ln -s net.lo net.eth0
rc-update add net.eth0 default
```

The hosts file

```
nano -w /etc/hosts
```

FILE /etc/hosts **Filling in the networking information**

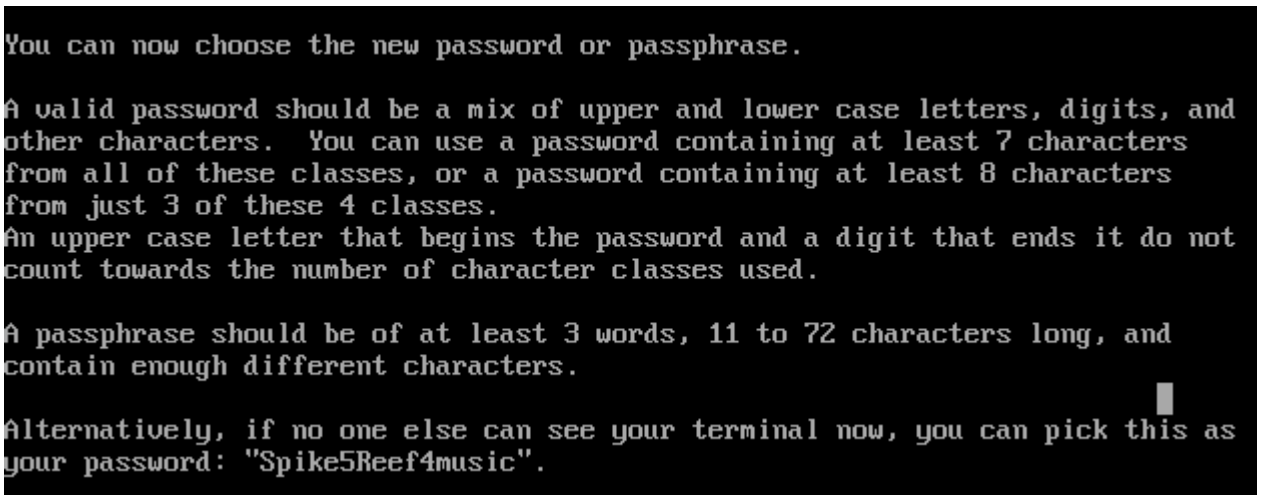
```
# This defines the current system and must be set
127.0.0.1     tux.homenetwork tux localhost

# Optional definition of extra systems on the network
192.168.0.5   jenny.homenetwork jenny
192.168.0.6   benny.homenetwork benny
```

System information

Root password

```
passwd
// enter a password
```



Init and boot configuration

OpenRC

```
nano -w /etc/rc.conf
nano -w /etc/conf.d/keymaps
nano -w /etc/conf.d/hwclock

// in all files change what's needed
```

systemd

```
systemd-firstboot --prompt --setup-machine-id
```

Installing system tools

System logger

```
emerge --ask app-admin/sysklogd
rc-update add sysklogd default
```

Filesystem tools

```
emerge --ask sys-fs/e2fsprogs
```

Networking tools

Installing a DHCP client

```
emerge --ask net-misc/dhcpcd
```

Configuring the bootloader

Selecting a boot loader GRUB2 (default)

Emerge

```
emerge --ask sys-boot/grub
// ensure GRUB_PLATFORMS="efi-64" is enabled using --verbose (and then remove it bef
```

Install

```
grub-install /dev/sda
// grub-install --target=x86_64-efi --efi-directory=/boot
```

Configure

```
grub-mkconfig -o /boot/grub/grub.cfg
```

Rebooting the system

```
exit

cd
umount -l /mnt/gentoo/dev{/shm,/pts,}
umount -R /mnt/gentoo
reboot
```

Finalizing

User administration

Adding a user for daily use

```
useradd -m -G users,wheel,audio -s /bin/bash <USERNAME>
passwd <USERNAME>
// type 2 times password
```

Disk cleanup

Removing tarballs

```
rm /stage3-*.tar.*
```

Genkernel