GentooInstallBattle2024

```
# 16TiB@512B by 32BitExt2(compat), 1EiB by 64bitExt4 ,other more (16EiB@4kB by 64BitBcachefs/Btrfs/Zfs)
dd if=/dev/zero of=ex.img bs=1 count=8192 seek=$(echo |awk '{ printf "%d",(2^31-1) }') ; #Calc by awk on 56Bit double-float presicion(compat)
sudo sgdisk --clear --zap-all ex.img ;
sudo sgdisk --set-alignment=4096 ex.img ;

# See https://github.com/uapi-group/specifications/blob/main/specs/discoverable_partitions_specification.md
# These UUID Specification would be revise each once every 9 months
# Example for x86/amd64 (root)
sudo sgdisk \
    --new=1:0:+300M \
    --typecode=1:024DEE41-33E7-11D3-9D69-0008C781F39F \
    --partition-guid=1:"$(uuidgen --time |cut -d'-' -f1-2)-$(uuidgen --random |cut -d'-' -f3-5)" \
    --change-name=1:"boot" \
    --new=2:0:+100M \
    --typecode=2:C12A7328-F81F-11D2-BA4B-00A0C93EC93B \
    --partition-guid=2:"$(uuidgen --time |cut -d'-' -f1-2)-$(uuidgen --random |cut -d'-' -f3-5)" \
    --change-name=2:"EFI System" \
    --new=3:0:0 \
    --typecode=3:4F68BCE3-E8CD-4DB1-96E7-FBCAF984B709 \
    --partition-guid=3:"$(uuidgen --time |cut -d'-' -f1-2)-$(uuidgen --random |cut -d'-' -f3-5)" \
    --change-name=3:"Linux filesystem" ex.img ;
sudo sgdisk -p ex.img ;
sudo kpartx -av ex.img ;

sudo kpartx -l ex.img ;

# 順にフォーマットする(Arch,Nixのカーネルデフォルトがext4のみなのを留意)
mkfs -t vfat -F 16 $(sudo kpartx -l ex.img |head -n1 |awk '{ print $1 }') ;
mkfs -t vfat -F 32 $(sudo kpartx -l ex.img |head -n2 |awk '{ print $1 }') ;
mkfs -t ext4 $(sudo kpartx -l ex.img |head -n3 |awk '{ print $1 }') ;
```

```
# MOUNT_POINT="/mnt/gentoo" ;

# With command grep, head and awk. See
https://naokton.hatenablog.com/entry/2021/07/23/094755

# mount $(sudo kpartx -l ex.img |head -n3 |awk '{ print $1 }') ${MOUNT_POINT} ;

MIRROR_SERVER="https://ftp.iij.ad.jp/pub/linux/gentoo" ;
STAGE3_TARBALL="${MIRROR_SERVER}/releases/amd64/autobuilds/$(curl -s
${MIRROR_SERVER}/releases/amd64/autobuilds/latest-stage3-amd64-systemd.txt |tail -n1
|cut -d' ' -f1)" ;

cd $(printf ${MOUNT_POINT} |rev |cut -d"/" -f2- |rev) && curl -fLO ${STAGE3_TARBALL} ; cd ;

cd ${MOUNT_POINT} && tar xpvf stage3-*.tar.xz --xattrs-include='*.*' --numeric-owner --keep-
newer-files ; cd ;

# cp -R /xxx/portage/* /etc/portage ;

# mount --types proc /proc ${MOUNT_POINT}/proc ;
# mount --rbind /sys ${MOUNT_POINT}/sys ;
# mount --make-rslave ${MOUNT_POINT}/sys ;
# mount --rbind /dev ${MOUNT_POINT}/dev ;
# mount --make-rslave ${MOUNT_POINT}/dev ;

mkdir -p ${MOUNT_POINT}/boot/efi ;

# See https://forum.linuxfoundation.org/discussion/857048/lfs201-chapter-17-discussion-
questions-banter-disk-partitioning
# mount $(sudo kpartx -l ex.img |head -n1 |awk '{ print $1 }') ${MOUNT_POINT}/boot ;
# mount $(sudo kpartx -l ex.img |head -n2 |awk '{ print $1 }') ${MOUNT_POINT}/boot/efi ;

# 実機: 場面により awk,dd,gptfdisk,util-linux(blkid) (必要最低限) 使う
# UEFI 向けに GPT でディスクをパーティショニングする(省略) {TGTDISK}=
{hda,vda,sda,nvme0n}
# (if need) mount /dev/${TGTDISK_PT1} /boot ;
# (if need) mount /dev/${TGTDISK_PT2} /boot/efi ;
```

```
# (必要ならchrootする) chroot ${MOUNT_POINT} /bin/bash ;

source /etc/profile ;

	# Sync official ebuild repository
	if which git 2> /dev/null ; then
		git clone https://github.com/gentoo-mirror/gentoo /var/db/repos/gentoo --depth
1 ;
	elif which rsync 2> /dev/null ; then
		emerge-webrsync ;
	else
		emerge --sync ;
	fi ;

eselect profile set --force $(eselect profile list |grep ".[0-9]/systemd" |sort -rV |cut -d' ' -f3 |tr -d
'[]' |head -n1) ;

	# Check rustc target
	if which rustc 2> /dev/null ; then
		rustc --print cfg -C target-cpu=native ;
	fi ;

	# Check the best CPU flag for the system
	# See
https://github.com/spack/spack/blob/develop/lib/spack/external/archspec/json/cpu/microarchite
ctures.json
	if which clang 2> /dev/null ; then
		cpu_flag=$(clang -march=native -E -v - </dev/null 2>&1 |grep cc1 |awk '{for(i=1;
i<=NF; i++) if($i~/target-cpu/) print $(i+1)}') ;
	elif which gcc 2> /dev/null ; then
		cpu_flag=$(gcc -march=native -E -v - </dev/null 2>&1 |grep cc1 |grep 'march'
|cut -d'=' -f2 |cut -d' ' -f1) ;
	else
		cpu_flag="generic" ;
	fi ; printf "CPU flag: ${cpu_flag}\n" ;

	case $(uname -m) ; in
		#
https://wiki.gentoo.org/wiki/Embedded_Handbook/Boards/StarFive_VisionFive_2
```

```
                # LicheeConsole / LicheePi / StarFive VisionFive
                "riscv64" ) COMMON_FLAGS="-march=rv64imafdc_zicsr_zba_zbb -
mtune=rocket -mabi=lp64d" ;;

                # 3A5000 https://zhuanlan.zhihu.com/p/39360027
                # fpu64 la64v100 https://loongson.github.io/LoongArch-
Documentation/LoongArch-toolchain-conventions-EN.html
                "loongarch64" ) COMMON_FLAGS="-march=loongarch64 -
mtune=loongarch64 -mabi=lp64d -msimd=lasx -mlasx" ;;

                # IBMPower10
                "powerpc64" ) COMMON_FLAGS="-march=power10 -mtune=power10 -
maltivec -mvsx -mvsx2 -mvsx3" ;;

                # IBMS390X(z16)
                "s390x" ) COMMON_FLAGS="-march=arch14 -mtune=arch14" ;;

                # AppleM2Ultra+ (but armv8 compatible for Google Tensor)
                "aarch64" ) COMMON_FLAGS="-march=armv8-
a+norcpc+crypto+simd+sha3+sve+sve2+bf16 -mtune=cortex-a77" ;;

                # Ryzen Threadripper PRO 7995WX
                "x86_64" ) COMMON_FLAGS="-march=x86-64-v3 -mtune=znver4 -mno-
3dnow -mno-sse4a" ;;

                # x86_64h(Apple,Rosetta2)
                "x86_64h" ) COMMON_FLAGS="-march=x86-64 -mtune=core-avx2 -mno-
sse4a" ;;

                # i686
                "i686" ) COMMON_FLAGS="-march=core2 -mtune=core2 -mno-3dnow -mno-
sse4a" ;;

                # else unknown other cpus
                "*" ) COMMON_FLAGS="-march=${cpu_flag} -mtune=native" ;;
        esac ; printf "COMMON_FLAGS: ${COMMON_FLAGS}\n" ;

cat << EOS > /etc/portage/make.conf
```

```
#EMERGE_DEFAULT_OPTS="-D4 --usepkg --usepkg-exclude 'app-portage/* app-crypt/* sys-kernel/* app-alternatives/* x11-drivers/* net-misc/inetutils' --buildpkg --buildpkg-exclude 'acct-user/* acct-group/* app-portage/* app-crypt/* */*-bin sys-kernel/* virtual/* app-alternatives/* x11-drivers/* net-misc/inetutils' --depclean-lib-check=y --implicit-system-deps=y --binpkg-respect-use=y --with-bdeps=y --tree -j5 -l4 --rebuilt-binaries"
EMERGE_DEFAULT_OPTS="-D4 --depclean-lib-check=y --implicit-system-deps=y --with-bdeps=y --tree -j5 -l4"
ACCEPT_LICENSE="* -@EULA -@OSI-APPROVED-NONFREE \
        rc dlj-1.1 FLEX man-pages unicode \
        public-domain freedist \
        linux-fw-redistributable \
        all-rights-reserved \
        unRAR Info-ZIP BZIP2 ZLIB \
        curl tcltk docbook SMAIL \
        libpng PCRE JSON \
        MSttfEULA PSF-2 \
        Intel-SDP \
        broadcom_bcm20702 Broadcom \
        NVIDIA-CUDA \
        AMD-GPU-PRO-EULA \
        microsoft-edge microsoft-vscode \
        Mojang \
        android google-chrome \
        metapackage"
PORTAGE_IONICE="chrt -r 99 ionice -c1 -n0 -p${PID}"
PORTAGE_NICENESS=-20
FEATURES="noinfo nodoc noman parallel-fetch parallel-install"
OBJCOPY="\${OBJCOPY} -g --strip-unneeded"
STRIP="\${STRIP} -s -S"
FCC=gfortran
FFLAGS="-g -O2"
MAKEOPTS="-j128"
COMMON_FLAGS="-O3 -pipe -w ${COMMON_FLAGS}"
CFLAGS="\${COMMON_FLAGS}"
CXXFLAGS="\${COMMON_FLAGS}"
EOS ;

yes "" |emerge -1 --keep-going --nodeps \
   sys-apps/portage \
```

```
    app-portage/gentoolkit \
    dev-vcs/git \
    sys-kernel/linux-headers \
    app-editors/vim \
    sys-process/htop \
    -X @installed ;

which clang 2> /dev/null && { cat << EOT >> /etc/portage/make.conf
CC=clang
CXX=clang++
CPP=clang-cpp
USE="cet sanitize clang systemd llvm-libunwind vaapi nvenc wayland sound-server -sdk -
sysprof -unwind"
EOT } ;

eselect kernel set 1 ;
eselect kernel set $(eselect binutils list |sort -rV |cut -d' ' -f2 |tr -d '[]' |head -n1) ;
eselect binutils set 1 ;
eselect binutils set $(eselect binutils list |sort -rV |cut -d' ' -f2 |tr -d '[]' |head -n1) ;
eselect gcc set 1 ;
eselect gcc set $(eselect gcc list |sort -rV |cut -d' ' -f2 |tr -d '[]' |head -n1) ;

source /etc/profile ;

emerge @preserved-rebuild ;
revdep-rebuild --ignore -p ;
yes "" |revdep-rebuild --ignore ;
yes "" |perl-cleaner --reallyall ;

rm -rf '/var/cache/binpkgs' ;

yes "" |emerge -uUN --keep-going @installed -X @world ;

eselect kernel set 1 ;
eselect kernel set $(eselect binutils list |sort -rV |cut -d' ' -f2 |tr -d '[]' |head -n1) ;
eselect binutils set 1 ;
eselect binutils set $(eselect binutils list |sort -rV |cut -d' ' -f2 |tr -d '[]' |head -n1) ;
eselect gcc set 1 ;
eselect gcc set $(eselect gcc list |sort -rV |cut -d' ' -f2 |tr -d '[]' |head -n1) ;
```

Linuxカーネルの導入(★)

(Alpine) apk add -l -u --clean-protected --allow-untrusted --force-overwrite --force-broken-world build-base make bzip2 xz pkgconf flex bison patch meson ninja perl python3 python3-dev py3-pip py3-numpy bash nano vim util-linux net-tools iproute2 htop btop iftop tcpdump procps iotop sysstat strace wget curl rsync git linux-headers libc-dev musl-dev musl gcompat zlib-static zstd-libs gettext-dev perl-dev glib-dev glib-static gcc g++ libgcc llvm lld clang libxml2-dev libxslt-dev libffi-dev readline-dev readline ncurses-libs wayland-dev mesa-dev openssl-dev cargo rust ;

(Arch) pacman -Sy --noconfirm make flex gcc bison ncurses elfutils perl bc openssl git zstd ;

(KDE neon/Ubuntu) apt -y install --force-yes --fix-broken --fix-missing build-essential bison flex libelf-dev bc libssl-dev git zstd ;

(RHEL) dnf -y install make flex gcc bison ncurses-devel elfutils-libelf-devel perl bc openssl-devel openssl-devel-engine git zstd ;

読む -> Building with LLVM https://docs.kernel.org/kbuild/llvm.html

(任意) 下記のmakeは make LLVM=1 と読み替える

mkdir /usr/src/$(uname -r |cut -d'-' -f1)-microsoft-standard-WSL2 ;

if [-f /usr/src/$(uname -r |cut -d'-' -f1)-microsoft-standard-WSL2] ; then

git clone https://github.com/microsoft/WSL2-Linux-Kernel.git /usr/src/$(uname -r |cut -d'-' -f1)-microsoft-standard-WSL2 --depth 1 ;

cd /usr/src/$(uname -r |cut -d'-' -f1)-microsoft-standard-WSL2 ;

(一部簡略) See popular distros: https://github.com/nyrahul/linux-kernel-configs

case $(uname -m) ; in

core/linux from loongarchlinux/core

"loongarch64")

REFKCONF_URI="https://raw.githubusercontent.com/loongarchlinux/core/refs/heads/main/linux/config.la64" ;;

core/linux-aarch64 from archlinuxarm/PKGBUILDs

"aarch64")

REFKCONF_URI="https://raw.githubusercontent.com/archlinuxarm/PKGBUILDs/refs/heads/master/core/linux-aarch64/config" ;;

packages/linux from archlinux/packaging (core)

```
                        "x86_64" )
REFKCONF_URI="https://gitlab.archlinux.org/archlinux/packaging/packages/linux/-/raw/main/config" ;;

                        # core/linux from archlinux32/packages
                        "i686" )
REFKCONF_URI="https://git.archlinux32.org/packages/plain/core/linux/config.pentium4" ;;
                esac ;

                curl -o ./.config ${REFKCONF_URI} ;
                yes "" |make oldconfig ;
                make -j${NPROC} && make modules_install -j${NPROC} && make install -j${NPROC} ;

                # (.wslconfig kernel=追記必要) 読めるようにWindowsのディレクトリに置く
                # cp ./vmlinux /mnt/c/WSL/Gentoo/vmlinux ;

                cd ;
        fi ;

# (好みで) Rustの導入を苦としない場合に、強力なCLIユーティリティを追加する
# emerge dev-lang/rust-bin ;
# [ -d /usr/lib/rustlib ] && cargo install bingrep broot chopstick difftastic duplink dysk eza erdtree ezr fclones fcp fd fend frs grex hck headtail huniq igrepper jless jql lsfp machin nomino normie ogrep pastel procs rargs rip ripgrep ripgrep-all rng-rename rnr sccache sshx termscp timetime timey trippy tuc unf unweave xcp xh ;
# cargo install cargo-update ;
# cargo install-update --all ;

# 日本語入力を追加する（Mozc）
emerge app-i18n/fcitx-mozc app-i18n/fcitx-configtool app-i18n/fcitx-gtk ;

# im-config -c ;
# /usr/libexec/mozc/mozc_tool --mode=config_dialog ;
# /usr/libexec/mozc/mozc_tool --mode=dictionary_tool ;
# /usr/libexec/mozc/mozc_tool --mode=word_register_dialog ;

cat << EOU >> /etc/environment
GTK_IM_MODULE=fcitx
```

```
QT_IM_MODULE=fcitx
XMODIFIERS=@im=fcitx
EOU ;

# ブートローダー(1か2) generic-ukiやnmbl等現時点では生えたばかりの概念すぎて今回は言及しない
# 1. systemd-boot(こっち)例
cat << EOV > /boot/loader/loader.conf
default gentoo
timeout 5
EOV ;
cat << EOW > /boot/loader/entries/gentoo.conf
title Gentoo Linux
linux /vmlinux
options root=PARTUUID=$(blkid -s PARTUUID -o value "$(sudo kpartx -l ex.img |head -n1 |awk '{ print $1 }')")
init=/usr/lib/systemd/systemd
EOW ;
bootctl --path=/boot install ;

# 2. GRUBのインストール(bootパーティション)
# cat << EOV >> /etc/portage/make.conf
# GRUB_PLATFORMS="efi-64 pc"
# EOV ;
# emerge sys-boot/grub ;

# grub-install --target=x86_64-efi --boot-directory=/boot --efi-directory=/boot/efi ;

# grub-mkconfig -o /boot/grub/grub.cfg ;

# テーマ例 https://github.com/Coopydood/HyperFluent-GRUB-Theme
# grub-mkfont --output=/boot/grub/fonts/dejavu24.pf2 --size=24 /usr/share/fonts/TTF/DejaVuSansMono.ttf ;
# grub.cfg に GRUB_FONT=/boot/grub/fonts/dejavu24.pf2 等追記

# (GUIデスクトップ) GNOME+ptyxis+gedit+firefox-binを導入
emerge --keep-going gnome-base/nautilus \
gnome-base/gnome-shell \
gnome-base/gnome-session \
```

```
gnome-base/gnome-control-center \
gnome-base/gnome-settings-daemon \
x11-terms/ptyxis \
app-editors/gedit \
www-client/firefox-bin ;

# umount -l ${MOUNT_POINT}/boot/efi ;
# umount -l ${MOUNT_POINT}/boot ;

# umount -l ${MOUNT_POINT}/dev ;
# umount -l ${MOUNT_POINT}/sys ;
# umount -l ${MOUNT_POINT}/proc ;

find /var/log -name *.log |xargs -n1 rm -rf ;

# 適切に校正されていない画像(に埋め込まれるヘッダ情報)を整理して圧縮
# (ただし最速のマシンでもルート全体にかける場合40分は掛かる)
# > mkarchisoでcustomize_airootfs.shがまだ有効なら
# > ↓突っ込んで自動化すると全体のディスクイメージサイズが25~53%減るかもしれない

# (find -size) avif -50M, svg(svgz) -8OM, xpm -100M, gif -290M
# AVIF: 'emerge media-libs/libavif ;'
# avifenc --codec aom --yuv 420 --min 20 --max 25 '{}' '{}'.avif ;

# gif <= Gifsicle
# emerge media-gfx/gifsicle ;
find ${MOUNT_POINT}/ -type f -size +2048 -size -200M -
iname '*.gif' -print0 |xargs -P 800 -n1 -0 gifsicle --
no-comments --lossy --use-colormap=web --optimize=3 --batch ;

# png <= OptiPNG
# emerge media-gfx/optipng ;
find ${MOUNT_POINT}/ -type f -size +2048 -size -1237M
iname '*.png' -print0 |xargs -P 800 -n1 -0 optipng -
snip -fO -o7 -zs1 -zm9 -strip all -preserve ;

# jpeg <= Jpegoptim
# emerge media-gfx/jpegoptim ;
find ${MOUNT_POINT}/ -type f -size +2048 -size -2575M -
```

```
iregex '.*\.\(jpg\|jpeg\|jpe\)$' -print0 |xargs -P 800
-n1 -0 jpegoptim --max=45 --all-progressive --strip-
all --preserve --totals --force ;

# tiffとbmpはIMで「一応圧縮通る(もはやそれだけの形骸物)」
# 他オプションはめんどくさい(どうにもならない)のでガン無視する
# emerge media-gfx/imagemagick ;

# tiff <= ImageMagick
find ${MOUNT_POINT}/ -type f -size +2048 -size -2653M -
iregex '*\.tiff?' -print0 |xargs -P 800 -n1 -0 magick
mogrify -format tiff -compress lzw ;

# bmp <= ImageMagick
find ${MOUNT_POINT}/ -type f -size +2048 -size -4100M
iname '*. bmp' -print0 |xargs -P 800 -n1 -0 magick
mogrify -format bmp -compress RLE ;

# (適当)確認してない (systemd-homed等将来レイアウト変わるとこのsort-index保証できてない、という意味)
# tarballを作成し、kpartx介した時のimgは万一やり直しの際バックアップも兼ねる
# cd ${MOUNT_POINT} && find . -mindepth 1 -maxdepth 1 |sort -rV -k2 -t'/' |sed -n -e '/\/opt/h' -e '/\/opt/d' -e '/\/bin/G;p' |sed -n -e '\/tmp/h' -e '\/tmp/d' -e '\/proc/G;p' |sed -n -e '\/root/h' -e '\/root/d' -e '\/etc/G;p' > $(pwd |rev |cut -d'/' -f2- |rev)/archive-gentoo.list && tar -I'zstd -v -T0 --zstd=strat=5,wlog=16,clog=9,hlog=11,slog=5,mml=6,tlen=8 --size-hint=8192' --one-file-system --acls --xattrs --numeric-owner -cSvpf $(pwd |rev |cut -d'/' -f2- |rev)/stage3-amd64-systemd-$(date +%Y%m%dT%H%M%SZ).tar.zst --files-from="$(pwd |rev |cut -d'/' -f2- |rev)/archive-gentoo.list" ; cd ;

sudo kpartx -dv ex.img ;

# Existing unused sparsed-hole, you should shrink from ex.img

# (export shrinked ex.img as ex-shrinked.img)
# truncate -s 0 ex-shrinked.img ;
# dd if=ex.img of=ex-shrinked.img bs=1 count=${SIZE} ;

# (export ex-shrinked.img as new-installmedia by direct)
# dd if=ex-shrinked.img of=/dev/sr0 bs=1 count=${SIZE} ;
```

フォント: 合わせたいのはEast Asian Width=大まかに「ギリシャキリル
+JP/KR/SC/TC/VN」が絶対
等幅(mono)なttfをDLして/usr/share/fonts下どこかにcpする
例. UDを /usr/share/fonts/noto/NotoSansMono-Regular.ttf に上書き
cp後 'sudo fc-cache -fv'

https://fonts.google.com/specimen/Reddit+Mono
https://github.com/yuru7/udev-gothic

(偏向な私見だが) ギリシャキリル文字が?なのでRedditと同じフォント、
JP/KR/SC/TC/VN(書体見本12文字+種字415字=32bitで収録可能な2万3059文字)
行政で使われる書体かつ可能な限りVSCode等で支障少ない漢字フォント(=UD?)

> East/Western Arabic(インドアラビア数字の系統とその地域文字群)等
> vteライブラリで対応(等幅で正確な表示がまだ現代でできていないか不明瞭)
> しているか怪しいものについてはここでは扱わないし言及しない
> リンク先はあくまで考え方として理解は難しいが共感はする
> Emacsフォント設定例 https://qiita.com/jado4810/items/5dd9f1b41ea5d1a4ec74