
#GentoolInstallBattleしませんか？



C89 東メ-35b

— 拝承製作所

1. まえがき

初めまして、(オズ) @Wizard_of_Oz__といます。この本を手にとって頂いたということはGentooもしくはGentooInstallBattleの事をご存知なのでしょうか？

GentooとはLinuxディストリビューションの一つであり、ソフトウェアの組み合わせを自分で選ぶことにより、痒いところに手が届く自分にとって最適な環境を作るためのディストリビューションです。

どのようなソフトウェアを入れ、どのような構成にするのかすべて自分の手で決めることができるのでカスタマイズや工作が好きな方にとってきっと気に入ることでしょう。

Gentooは環境を細かくカスタマイズするために、すべてのソフトウェアを自分でコンパイルします。あなたがEmacsを使用しているとして、X上でEmacsが起動する必要があるのでしょうか？GentooではUSEフラグというオプションを選択することで、必要な機能のみに絞ってインストールすることができます。

ここでGentooのインストールハンドブックからの言葉を引用します。

Gentooでは、全てが選択です。

また、このようにも書かれています。

Gentooに実行させることの選択肢を全員が理解していることは、きわめて重要です。

Gentooでは全てを選択できます。つまり全て何を選ぶか決める必要があります。Ubuntuなどでは大多数の人が困らないような設定がされています。そのため誰でも気軽に使い始めることができますが、自分にとって気に入らない設定を変えることは難しくなっています。Gentooはその逆で、気に入らない設定があれば全てを変更することができます。(プロプライエタリ以外の)どのようなソフトウェアもソースコードからコンパイルし、いくらでも改変できるような環境が整っています。その代わり、Gentooはインストール時から常にユーザに選択を迫ってきます。今までカーネルのオプションを気にしたことはありますか？Gentooはインストール時にカーネルコンパイルを行う必要があります。

Gentooは素晴らしいディストリビューションなのですが、上記のように気軽に使ってもらうことが難しく布教し辛い側面がありました。そこで、まずはVM上で#GentooInstallBattleをおこない、Gentooの楽しさを知って新たなユーザーとなって貰えたらと思います。

#GentooInstallBattleとはtwitter片手に行われるエクストリーム・スポーツです。

#gentooinstallbattle は Gentoo(OSは問わない)のインストールをこの hash tag つけて実況しながらインストールするだけの簡単なバトルだよ! 誰とのバトル? 同じhashtagをつけた強敵(とも)や自分自身...マシンや.....カーネル設定とのバトルさ!

— @naota344

現代ではGentoo上にmikutterというLinuxでのデファクトスタンダードとなったtwitterクライアントを インストールしてつぶやけば終了というルールになっています。

公式のハンドブックを片手に#GentooInstallBattleに立ち向かっていくのも楽しいですが、情報が多く判断に迷う場面が多々出てくると思います。そこで、インストール手順の一例を示しGentooに触れるきっかけにして頂きたいと思います。この手順に従えば、何も選択しなくてもインストール完了できると思います。その後、気になるところを自分の手で少しずつ変えていくなどしてGentooの世界を楽しんでください。

2. 事前準備

下記手順はWindows上でVMware Playerを使用する環境を想定しています。しかしVMwareに特化した設定を除き、直接インストールする場合も特に違いはないかと思います。まずはVMware Playerを入手してインストールしてください。VMware Playerのインストールについては本筋ではないのでここでは割愛します。

次に、Gentooのインストールメディアをダウンロードしましょう。Gentoo公式ページ(<https://www.gentoo.org/>)に行き、Downloadsタブを選んでください。大きく64ビット用のamd64と32ビット用のx86に分かれています。本書はamd64を使用することにします。Boot Mediaの中にMinimal Installation CDとHybrid ISO(LiveDVD)があります。ここではMinimal Installation CDを使用しますが、物理環境にインストールする場合は Hybrid ISO(LiveDVD)を使用するとXを起動してブラウザでハンドブックを見ながら構築ができるので良いかと思います。Stage Archivesは後ほど別の方法で入手するのでここではダウンロードする必要はありません。

インストールイメージが入手できたらまずはVMの作成を行います。仮想マシン作成時に注意することは、ネットワークとプロセッサ・メモリの設定です。

Gentooインストール時にインターネットの接続は不可欠です。自分の環境に合わせて設定してください。また後述しますがインストール作業はVMの外からsshで入り操作することを推奨します。外から繋がられるようネットワークアダプタはNATではなくブリッジが良いかと思います。

Gentooではインストール時から大量にコンパイルを行います。 プロセッサやメモリは与えられるだけ与えておいたほうが無駄に待たずに済みます。

VMを構築し、ダウンロードしたインストールメディアのISOを接続したらVMを起動しましょう。

3. boot

VMを起動したらまずはCDからのブートプロンプトが表示されるので、Enterを入力して起動させます。 VMの画面をクリックしてコントロールを与えておかないと入力を受け付けないので注意。 Enterを押さず15秒経過するとデフォルトのブートに進んでしまう(VMの場合PXEが走るはず)ので、 その場合は再起動して再度ブートプロンプトを表示させましょう。

少し待つとキーマップを入力する画面となります。 放っておくとUS配列として認識して進んでしまうので、日本語キーボードを使用している人は ここで22 Enter と入力しましょう。

しばらくするとブートが完了し、自動的にrootとしてログインされた状態となります。 まずはNetの接続が正しくできているか確認しましょう。 VMの設定が正しくされていれば、以下のように外部への接続が確認できるはずです。

```
# ping -c 1 google.co.jp
PING google.co.jp (216.58.220.195) 56(84) bytes of data.
64 bytes from nrt13s36-in-f3.1e100.net (216.58.220.195): icmp_seq=1 ttl=54
time=21.3 ms
```

ifconfigコマンドはobsoleteなのでip aコマンドでIPアドレスの設定を確認します。

```
#ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 brd 127.255.255.255 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eno16777736: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UP group default qlen 1000
    link/ether aa:bb:cc:dd:ee:ff brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.1/24 brd 192.168.0.255 scope global enp3s0
        valid_lft forever preferred_lft forever
    inet6 fe80::be5f:f4ff:fe20:44aa/64 scope link
        valid_lft forever preferred_lft forever
```

eno<数字>のようになっているのがEtherのポートです。 VM上で操作するとマウスの制御が取られたままでWebなどを見ながらの操作がやりにくいので、 IPアドレスが確認できたあとはsshでいつもの環境から操作することにする。

sshで接続するためにはまずrootのパスワードを設定する必要があります。 ちなみにここで設定するパスワードは一時的に使用されるもので、 インストールするOSにはまた後ほど設定する必要がある。

```
# passwd
New password: <hogebuga>
Retype new password: <hogebuga>
passwd: password updated successfully
```

その後、sshdを起動させる

```
# /etc/init.d/sshd start
```

ここまで出来たらVMからフォーカスを外してTeraTermなりLinuxからsshする。 VMからフォーカスを外す方法はCtrl+Alt同時押し。 ip aコマンドで確認した確認したIPアドレスにsshします。 IDはroot、Passwordは上記で設定した<hogebuga>となる。

4. パーティション分け

パーティションを設定する際に、まず最初にMBR、GPTどちらを使うか選択する必要がある。 GPTのほうが新しく、2T以上のパーティション作成やUEFIブートをするためにはGPTが必要となる。 通常はGPTで問題ないでしょう。

Gentooハンドブックとは異なり下記構成のパーティションを作成する。 1つ目のパーティションはブートローダー用で、 GPTでgrubを使用する場合、先頭に"BIOS boot partition"というフラグを立てる必要がある。 2つ目のパーティションはブート用の領域で、カーネルイメージなどが格納される。 ハンドブックでは128MBだが、安全のため古いカーネルを残しておくとすぐ溢れてしまうため256Mとしている。 3つ目のパーティションはルート用であり、残り全ての容量を割り当てる。 今回はVMでHDDの追加なども容易であるので以下のようにしているが、 物理環境では/homeを別パーティションにするなど検討したほうが良いと思う。

Swapパーティションが無いことに疑問を持つかもしれないが、 Swapはファイルを使用しても性能に特に差はなく(btrfsを使用する場合を除く)、 VMの場合HDDの使用容量が固定されないファイルのほうが適していると思い以下のようにしている。 Swapファイルの設定方

法は後ろのほうで説明するが、メモリ容量の少ない環境の場合はパーティション作成後に実施したほうが良いかもしれない。

```
Partition Filesystem Size Description
/dev/sda1 (bootloader) 2M BIOS boot partition
/dev/sda2 ext2 256M Boot partition
/dev/sda3 ext4 残り全て Root partition
```

GPTの操作としてGentoo Handbookではpartedの使用例が記載されているが、partedは操作が即時反映されてしまったり、容量を指定したパーティション作成ができないなどの問題がある。そこで、ここではgdiskというGPT用のパーティショニングツールを使用する。ちなみに、以前はfdiskはGPT対応していなかったが、現在では問題なく使用できるのでfdiskに慣れている人はそちらを使用しても問題ない。

#gdisk <デバイス名>で起動する。

```
livecd ~ # gdisk /dev/sda
GPT fdisk (gdisk) version 0.8.10

Partition table scan:
MBR: not present
BSD: not present
APM: not present
GPT: not present

Creating new GPT entries.
```

?を入力するとコマンドのヘルプが出る。

```
Command (? for help): ?
b      back up GPT data to a file
c      change a partition's name
d      delete a partition
i      show detailed information on a partition
l      list known partition types
n      add a new partition
o      create a new empty GUID partition table (GPT)
p      print the partition table
q      quit without saving changes
r      recovery and transformation options (experts only)
s      sort partitions
t      change a partition's type code
v      verify disk
w      write table to disk and exit
```

```
x      extra functionality (experts only)
?      print this menu
```

pを入力すると書き込む予定のパーティション情報が表示される。

```
Command (? for help): p
Disk /dev/sda: 16777216 sectors, 8.0 GiB
Logical sector size: 512 bytes
Disk identifier (GUID): 382984F4-40A4-43DB-A363-80E723BB7489
Partition table holds up to 128 entries
First usable sector is 34, last usable sector is 16777182
Partitions will be aligned on 2048-sector boundaries
Total free space is 16777149 sectors (8.0 GiB)
```

Number	Start (sector)	End (sector)	Size	Code	Name
--------	----------------	--------------	------	------	------

nコマンドでパーティションを作成する。 <>で囲んだ部分がユーザの入力となる。 入力が必要な項目は容量ぐらいで、その他の部分はデフォルトで問題ないよう設定される。 1パーティション目はBIOS boot partitionを設定しなければいけないので 一度Lを入力しコードを確認している。

```
Command (? for help): <n>
Partition number (1-128, default 1): <Enter>
First sector (34-16777182, default = 2048) or {+-}size{KMGTP}: <Enter>
Last sector (2048-16777182, default = 16777182) or {+-}size{KMGTP}: <+2M>
Current type is 'Linux filesystem'
Hex code or GUID (L to show codes, Enter = 8300): <l>
0700 Microsoft basic data  0c01 Microsoft reserved  2700 windows RE
3000 ONIE boot              3001 ONIE config          4100 PowerPC PreP boot
4200 windows LDM data       4201 windows LDM metadata 7501 IBM GPFS
7f00 ChromeOS kernel       7f01 ChromeOS root        7f02 ChromeOS reserved
8200 Linux swap             8300 Linux filesystem      8301 Linux reserved
8302 Linux /home            8400 Intel Rapid Start    8e00 Linux LVM
a500 FreeBSD disklabel     a501 FreeBSD boot         a502 FreeBSD swap
a503 FreeBSD UFS            a504 FreeBSD ZFS          a505 FreeBSD Vinum/RAID
a580 Midnight BSD data     a581 Midnight BSD boot    a582 Midnight BSD swap
a583 Midnight BSD UFS      a584 Midnight BSD ZFS     a585 Midnight BSD Vinum
a800 Apple UFS              a901 NetBSD swap          a902 NetBSD FFS
a903 NetBSD LFS             a904 NetBSD concatenated  a905 NetBSD encrypted
a906 NetBSD RAID            ab00 Apple boot           af00 Apple HFS/HFS+
af01 Apple RAID             af02 Apple RAID offline   af03 Apple label
af04 AppleTV recovery       af05 Apple Core Storage   be00 Solaris boot
bf00 Solaris root           bf01 Solaris /usr & Mac z  bf02 Solaris swap
bf03 Solaris backup         bf04 Solaris /var         bf05 Solaris /home
bf06 Solaris alternate se  bf07 Solaris Reserved 1   bf08 Solaris Reserved 2
```

```
bf09 Solaris Reserved 3    bf0a Solaris Reserved 4    bf0b Solaris Reserved 5
c001 HP-UX data            c002 HP-UX service        ea00 Freedesktop $BOOT
eb00 Haiku BFS             ed00 Sony system partiti  ed01 Lenovo system partit
ef00 EFI System           ef01 MBR partition scheme ef02 BIOS boot partition
fb00 VMware VMFS          fb01 VMware reserved      fc00 VMware kcore crash p
fd00 Linux RAID
Hex code or GUID (L to show codes, Enter = 8300): <ef02>
Changed type of partition to 'BIOS boot partition'
```

同様に2パーティション目を作成する。ここでは容量のみ指定し、ほかはデフォルト(エンター)

```
Command (? for help): <n>
Partition number (2-128, default 2): <Enter>
First sector (34-16777182, default = 6144) or {+-}size{KMGTP}: <Enter>
Last sector (6144-16777182, default = 16777182) or {+-}size{KMGTP}: <+256M>
Current type is 'Linux filesystem'
Hex code or GUID (L to show codes, Enter = 8300): <Enter>
Changed type of partition to 'Linux filesystem'
```

3パーティション目は残り全ての容量を割り当てるので、全てエンターでOK。

```
Command (? for help): <n>
Partition number (3-128, default 3): <Enter>
First sector (34-16777182, default = 530432) or {+-}size{KMGTP}: <Enter>
Last sector (530432-16777182, default = 16777182) or {+-}size{KMGTP}: <Enter>
Current type is 'Linux filesystem'
Hex code or GUID (L to show codes, Enter = 8300): <Enter>
Changed type of partition to 'Linux filesystem'
```

pコマンドで設定した情報を確認する。

```
Command (? for help): p
Disk /dev/sda: 16777216 sectors, 8.0 GiB
Logical sector size: 512 bytes
Disk identifier (GUID): 382984F4-40A4-43DB-A363-80E723BB7489
Partition table holds up to 128 entries
First usable sector is 34, last usable sector is 16777182
Partitions will be aligned on 2048-sector boundaries
Total free space is 2014 sectors (1007.0 kiB)
```

Number	Start (sector)	End (sector)	Size	Code	Name
1	2048	6143	2.0 MiB	EF02	BIOS boot partition
2	6144	530431	256.0 MiB	8300	Linux filesystem

3	530432	16777182	7.7 GiB	8300	Linux filesystem
---	--------	----------	---------	------	------------------

確認し問題なければいよいよHDDに書き込む。 問題があればqコマンドで何も変更せず終えることができる。 物理環境で作業する場合、ここで一息ついてよく確認してから wコマンドを実行し書き込みましょう。

```
Command (? for help): <w>
```

```
Final checks complete. About to write GPT data. THIS WILL OVERWRITE EXISTING PARTITIONS!!
```

```
Do you want to proceed? (Y/N): <y>
```

```
OK; writing new GUID partition table (GPT) to /dev/sda.
```

```
The operation has completed successfully.
```

書き込みが完了するとgdiskが終了します。 下記コマンドで現在のHDD情報が確認できます。

```
lived ~ # gdisk -l /dev/sda
GPT fdisk (gdisk) version 0.8.10
```

```
Partition table scan:
```

```
MBR: protective
```

```
BSD: not present
```

```
APM: not present
```

```
GPT: present
```

```
Found valid GPT with protective MBR; using GPT.
```

```
Disk /dev/sda: 16777216 sectors, 8.0 GiB
```

```
Logical sector size: 512 bytes
```

```
Disk identifier (GUID): 382984F4-40A4-43DB-A363-80E723BB7489
```

```
Partition table holds up to 128 entries
```

```
First usable sector is 34, last usable sector is 16777182
```

```
Partitions will be aligned on 2048-sector boundaries
```

```
Total free space is 2014 sectors (1007.0 kiB)
```

Number	Start (sector)	End (sector)	Size	Code	Name
1	2048	6143	2.0 MiB	EF02	BIOS boot partition
2	6144	530431	256.0 MiB	8300	Linux filesystem
3	530432	16777182	7.7 GiB	8300	Linux filesystem

***old

MBRでは従来fdiskコマンドを使用していたが、
GPTでフォーマットする場合partedコマンドを使用する

```
# parted -a optimal /dev/sda
```

-a optimalオプションはパーティション分割の際のアライメントを適切に設定してくれるもの

```
lived ~ # parted -a optimal /dev/sda
GNU Parted 3.2
Using /dev/sda
welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
```

ここでhelpと打てばコマンドの一覧が表示される
またprintと打てば現在のパーティション情報が表示される

まずはHDDにGPTであることを設定する
この操作を行うことで既存のデータは全て消えてしまうので
もし既存の環境に対して行う場合は注意が必要である

```
(parted) mklabel gpt
```

次にBIOS boot partitionを作成する

まず、以下のコマンドで容量の操作単位をMiB(メビバイト)にする

```
(parted) unit mib
```

以下の操作で2Mのパーティションを作成する

これはHDD先頭から数えて1MiB〜3MiB(2MiB分)の区間を1パーティションにするという意味になる

```
(parted) mkpart primary 1 3
```

作成したパーティションにgrubという名前をつけて、bios_grubオプションを立てる

GPTパーティションでgrubを使うにはこのbios_grubオプションが必要になるようだ

```
(parted) name 1 grub
```

```
(parted) set 1 bios_grub on
```

```
(parted) print
```

Model: VMware, VMware Virtual S (scsi)

Disk /dev/sda: 20480MiB

Sector size (logical/physical): 512B/512B

Partition Table: gpt

Disk Flags:

Number	Start	End	Size	File system	Name	Flags
1	1.00MiB	3.00MiB	2.00MiB		grub	bios_grub

パーティションの作成が終わったら、次はフォーマットを行う。 以前に記載した通り、bootパーティションはext2、rootパーティションはext4でフォーマットする。

```
livecd ~ # mkfs.ext2 /dev/sda2
mke2fs 1.42.13 (17-May-2015)
Creating filesystem with 262144 1k blocks and 65536 inodes
Filesystem UUID: 7197f997-5c15-4c44-b5d6-e060b4206bc3
Superblock backups stored on blocks:
8193, 24577, 40961, 57345, 73729, 204801, 221185

Allocating group tables: done
writing inode tables: done
writing superblocks and filesystem accounting information: done
```

```
livecd ~ # mkfs.ext4 /dev/sda3
mke2fs 1.42.13 (17-May-2015)
Creating filesystem with 2030843 4k blocks and 507904 inodes
Filesystem UUID: 47687742-0e76-443d-94c2-941e852bb12b
Superblock backups stored on blocks:
32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

5. ベースシステムのインストール

ここから本格的にインストール作業に入っていくが、 インストールを始める前に、日時が正しいか確認する

```
livecd ~ # date
Sat Dec 26 11:40:26 UTC 2015
```

VMで構築している場合、時刻はあっているように見えても実はUCT時間なので異なっている。時刻を合わせる手っ取り早い方法は下記コマンドでNTPを使用することである。

```
# ntpdate ntp.nict.jp
```

まずはrootパーティションをマウントし、 その中にbootディレクトリを作成してbootパーティションをマウントする。

```
livecd ~ # mount /dev/sda3 /mnt/gentoo/  
livecd ~ # mkdir /mnt/gentoo/boot  
livecd ~ # mount /dev/sda2 /mnt/gentoo/boot/
```

HDDの準備ができたので、次からGentooのインストールを始める。まず最初にインストール先に移動する。

```
livecd ~ # cd /mnt/gentoo/
```

システムのベースとなるstage tarball(stage3)をダウンロードする。64ビット版の最新は下記URLに格納される。

```
http://ftp.jaist.ac.jp/pub/Linux/Gentoo/releases/amd64/autobuilds/current-stage3-amd64/
```

linksというテキストブラウザが使用できるので、以下のようにして起動する。

```
livecd ~ # links http://ftp.jaist.ac.jp/pub/Linux/Gentoo/releases/amd64/autobuilds/  
current-stage3-amd64/
```

上下キーでカーソルが移動し、ダウンロードしたいファイルのファイル名上でエンターを押せば保存するか聞かれる。またlinksの終了方法はQキーを押す。

ファイル名がわかっている場合、以下のように直接ファイルを確認・指定してダウンロードする方法もある。

```
livecd gentoo # wget http://ftp.jaist.ac.jp/pub/Linux/Gentoo/releases/amd64/  
autobuilds/current-stage3-amd64/stage3-amd64-20151227.tar.bz2  
  
--2015-12-30 00:23:41-- http://ftp.jaist.ac.jp/pub/Linux/Gentoo/releases/amd64/  
autobuilds/current-stage3-amd64/stage3-amd64-20151227.tar.bz2  
Resolving ftp.jaist.ac.jp... 150.65.7.130, 2001:df0:2ed:feed::feed  
Connecting to ftp.jaist.ac.jp|150.65.7.130|:80... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 249998079 (238M) [application/x-bzip2]  
Saving to: 'stage3-amd64-20151227.tar.bz2'  
  
stage3-amd64-20151227.tar 100%[=====>] 238.42M  
3.08MB/s in 79s  
  
2015-12-30 00:25:00 (3.02 MB/s) - 'stage3-amd64-20151227.tar.bz2' saved  
[249998079/249998079]
```

ダウンロードしたファイルが壊れていないかどうかを確認するためには、以下のファイルもダウンロードしてください。

```
stage3-amd64-20151227.tar.bz2.DIGESTS
```

このファイルにはstage3とstage3のCONTENTS(ファイルの内容一覧)のチェックサムが記載されています。以下のコマンドでSHA512を計算することができますが、この長い数列を目視で確認するのは大変です。

```
livecd gentoo # sha512sum stage3-amd64-20151227.tar.bz2
b4708fbdc6acff3276dcc89c063a9bbbed142b98eddfba2637b47f5fa73e93d6ce9c1f1015a824d4910852f338e6e
stage3-amd64-20151227.tar.bz2
```

以下のコマンドを実行すると、SHA512を計算して一致したか結果を表示してくれます。

```
livecd gentoo # sha512sum -c stage3-amd64-20151227.tar.bz2.DIGESTS
stage3-amd64-20151227.tar.bz2: OK
stage3-amd64-20151227.tar.bz2: FAILED
sha512sum: stage3-amd64-20151227.tar.bz2.CONTENTS: No such file or directory
stage3-amd64-20151227.tar.bz2.CONTENTS: FAILED open or read
sha512sum: stage3-amd64-20151227.tar.bz2.CONTENTS: No such file or directory
stage3-amd64-20151227.tar.bz2.CONTENTS: FAILED open or read
sha512sum: WARNING: 2 listed files could not be read
sha512sum: WARNING: 1 computed checksum did NOT match
```

2行目のFAILEDは気にしなくても大丈夫です。1行目(とCONTENTSもダウンロードした場合は3行目)がOKとなっていればファイルは正常です。

ダウンロードが完了したら、stage3を展開します。

```
livecd gentoo # tar xvjpf stage3-amd64-20151225.tar.bz2
...
./sbin/nologin
./sbin/ip6tables-restore
./sbin/consoletype
livecd gentoo #
```

展開が終わったらコンパイルオプションを設定します。

```
#nano -w etc/portage/make.conf
```

CPUに最適化するため、CFLAGSに-march=nativeを追加します。コンパイル時に使用するコア数を設定するためMAKEOPTS="-j<core数>"を追加します。

```
CFLAGS="-O2 -pipe -march=native"
MAKEOPTS="-j1"
```

Ctrl+xで保存します。

DNS情報をコピーします。

```
livedd gentoo # cp -L /etc/resolv.conf etc/
```

これでベースシステムのインストールが終わりました。次はインストール先の環境に入り、システムをインストールしていきます。

6. Gentoo環境の設定

ベースシステムがインストールできたので、インストールした環境に入り 残りのシステムをインストールします。chrootコマンドでインストールした環境に入るのですが、環境に入る前に必要なファイルシステムをマウントします。

```
livedd gentoo # mount -t proc proc proc/
livedd gentoo # mount --rbind /sys sys/
livedd gentoo # mount --rbind /dev dev/
```

マウントしたらついにインストールした新しい環境に入ります。

```
livedd gentoo # chroot /mnt/gentoo/ /bin/bash
livedd / # source /etc/profile
livedd / # export PS1=''(chroot) $PS1''
```

まず下記コマンドでPortageを使用できるようにしましょう。

```
(chroot) livedd / # emerge-webrsync
!!! Section 'gentoo' in repos.conf has location attribute set to nonexistent
    directory: '/usr/portage'
!!! Section 'x-portage' in repos.conf has location attribute set to nonexistent
    directory: '/usr/portage'
!!! Invalid Repository Location (not a dir): '/usr/portage'
Fetching most recent snapshot ...
```

```
Trying to retrieve 20151225 snapshot from http://distfiles.gentoo.org ...
Fetching file portage-20151225.tar.xz.md5sum ...
Fetching file portage-20151225.tar.xz.gpgsig ...
Fetching file portage-20151225.tar.xz ...
Checking digest ...
Getting snapshot timestamp ...
Syncing local tree ...

Number of files: 206,664 (reg: 179,479, dir: 27,185)
Number of created files: 206,663 (reg: 179,479, dir: 27,184)
Number of deleted files: 0
Number of regular files transferred: 179,479
Total file size: 394.73M bytes
Total transferred file size: 394.73M bytes
Literal data: 394.73M bytes
Matched data: 0 bytes
File list size: 5.18M
File list generation time: 0.001 seconds
File list transfer time: 0.000 seconds
Total bytes sent: 175.63M
Total bytes received: 3.54M

sent 175.63M bytes  received 3.54M bytes  114.45K bytes/sec
total size is 394.73M  speedup is 2.20
Cleaning up ...

* IMPORTANT: 9 news items need reading for repository 'gentoo'.
* Use eselect news read to view new items.
```

タイムゾーンを設定します。

```
(chroot) livedcd zoneinfo # echo "Japan" > /etc/timezone
(chroot) livedcd zoneinfo # emerge --config sys-libs/timezone-data
Configuring pkg...
* updating /etc/localtime with /usr/share/zoneinfo/Japan
```

ロケールを設定します。使用するロケールのみコメントを外します。英語と日本語の環境2つがあればいたい事足りるでしょう。

```
(chroot) livedcd / # nano -w /etc/locale.gen

en_US.UTF-8 UTF-8
ja_JP.UTF-8 UTF-8
```

ファイルを編集したら設定します。

```
(chroot) livedcd / # locale-gen
* Generating 2 locales (this might take a while) with 1 jobs
* (1/2) Generating en_US.UTF-8 ...
      [ ok ]
* (2/2) Generating ja_JP.UTF-8 ...
      [ ok ]
* Generation complete
```

設定したロケールを確認します。

```
(chroot) livedcd / # eselect locale list
Available targets for the LANG variable:
[1]   C
[2]   POSIX
[3]   en_US.utf8
[4]   ja_JP.utf8
[ ]   (free form)
```

どのロケールを使用するか選択します。

```
(ch) livedcd / # eselect locale set 4
Setting LANG to ja_JP.utf8 ...
Run ``. /etc/profile'' to update the variable in your shell.
```

ここで/etc/profileをアップデートすると言われるので従いましょう。プロンプトの設定も初期化されるので再度設定します。

```
(chroot) livedcd / # env-update && source /etc/profile
>>> Regenerating /etc/ld.so.cache...
livedcd / # export PS1='(chroot) $PS1'
(chroot) livedcd / #
```

7. Linux Kernelの設定

いよいよカーネルの設定です。ただし今回は手動で設定しません。 インストールイメージと同じ設定をそのまま流用します。

まずはカーネルソースをインストールします。

```
(chroot) livedcd / # emerge --ask sys-kernel/gentoo-sources
```



```
* IMPORTANT: 9 news items need reading for repository 'gentoo'.
* Use eselect news read to view new items.
```

These are the packages that would be merged, in order:

Calculating dependencies... done!

```
[ebuild N      ] sys-devel/bc-1.06.95-r1 USE='readline -libedit -static'
[ebuild N      ] sys-kernel/gentoo-sources-4.1.12 USE='-build -experimental -
kdbus -symlink'
```

would you like to merge these packages? [Yes/No]

インストールされているカーネルソースを確認します。 現在はひとつだけですが、複数ある場合は使用するカーネルソースを選択します。

```
(chroot) livedd / # eselect kernel list
Available kernel symlink targets:
[1]  linux-4.1.12-gentoo *
```

カーネルソースの場所に移動します。 上記で選択したカーネルが/usr/src/linuxにリンクされています。

```
(chroot) livedd / # ls -l /usr/src/
合計 4
lrwxrwxrwx  1 root root   19 12月 27 08:54 linux -> linux-4.1.12-gentoo
drwxr-xr-x 24 root root 4096 12月 27 08:54 linux-4.1.12-gentoo
(chroot) livedd / # cd /usr/src/linux
```

ここでカーネルの設定を行います。 手動で設定する場合、make menuconfigコマンドを使用するのですが、今回はmake localyesconfigコマンドを使用しています。認識しているものをそのまま引き継ぎます。 コマンドを実行すると、新しく追加された設定についていくつも質問されます。 一つ一つ調べて設定するのは大変なので、全てエンターで切り抜けます。特に問題はないはずです。

```
(chroot) livedd linux # make localyesconfig
.
.
.
Support non-standard NVDIMMs and ADR protected memory (X86_PMEM_LEGACY) [N/y/?]
(NEW)
MultiProtocol Label Switching (MPLS) [N/y/?] (NEW)
Persistent memory block device support (BLK_DEV_PMEM) [N/m/y/?] (NEW)
```

```
request-based DM: use blk-mq I/O path by default (DM_MQ_DEFAULT) [N/y/?] (NEW)
Log writes target support (DM_LOG_WRITES) [N/m/y/?] (NEW)
Cadence devices (NET_CADENCE) [Y/n/?] (NEW)
```

```
#
# configuration written to .config
#
```

カーネルの設定が完了したら、ついにカーネルコンパイルとインストールです。ここで-j<コア数> オプションを指定します。上記で設定したものは Portage にしか適用されないためです。

```
(chroot) livedd linux # make -j2
(chroot) livedd linux # make modules_install
(chroot) livedd linux # make install
```

ファイルのマウントを設定します。ひな形があるので少し修正するだけです。

```
nano -w /etc/fstab
# NOTE: If your BOOT partition is ReiserFS, add the notail option to opts.
/dev/BOOT          /boot             ext2              noauto,noatime   1 2
/dev/ROOT          /                 ext3              noatime           0 1
/dev/SWAP          none              swap              sw                0 0
/dev/cdrom         /mnt/cdrom        auto              noauto,ro        0 0
/dev/fd0           /mnt/floppy       auto              noauto           0 0
```

BOOTをsda2に、ROOTをsda3に変更します。SWAPは作成していないのでコメントアウトします。フロッピーを使用しない場合(しませんよね?)fd0もコメントアウトします。/bootに記載されているnoautoオプションは削除するのがお勧めです。カーネル更新時にマウントし忘れてトラブルになるのを防ぎます。

```
# NOTE: If your BOOT partition is ReiserFS, add the notail option to opts.
/dev/sda2          /boot             ext2              noatime           1 2
/dev/sda3          /                 ext4              noatime           0 1
#/dev/SWAP         none              swap              sw                0 0
/dev/cdrom         /mnt/cdrom        auto              noauto,ro        0 0
#/dev/fd0          /mnt/floppy       auto              noauto           0 0
```

ホストネームの設定をしましょう。下記ファイルに書き込みます。この例ではgentooと設定しています。

```
nano -w /etc/conf.d/hostname
```

```
hostname="gentoo"
```

DNSを使用していない場合、下記ファイルから.\0を削除したほうがログイン画面で余計な表示がなくなります。

```
nano -w /etc/issue
This is \n.\0 (\s \m \r) \t
```

```
This is \n (\s \m \r) \t
```

ネットワーク設定を行います。まずは下記を実行。

```
(chroot) livedd / # emerge --ask --noreplace net-misc/netifrc
```

IPの設定を行います。DHCPを使用する場合、下記のように記載します。

```
(chroot) livedd / # nano /etc/conf.d/net
config_eno16777736='dhcp'
```

起動時にネットが繋がるよう設定します。

```
(chroot) livedd / # cd /etc/init.d/
(chroot) livedd init.d # ln -s net.lo net.eno16777736
(chroot) livedd init.d # rc-update add net.eno16777736 default
* service net.eno16777736 added to runlevel default
```

システムのrootパスワードを設定します。これは最初に設定したものと別のものでも構いません。下記例はgentooと設定していて、簡単なので怒られています(が、設定はできています)。

```
(chroot) livedd / # passwd
新しいパスワード:
よくないパスワード: 辞書の単語に基づいています
よくないパスワード: 簡単すぎます
新しいパスワードを再入力してください:
passwd: パスワードは正しく更新されました
```

キーボードの設定をしましょう。デフォルトではUS配列のため、日本語キーを使用している場合はkeymapの項目を修正します。

```
nano -w /etc/conf.d/keymaps
keymap='us'
```

```
keymap='jp106'
```

///etc/conf.d/hwclock

そろそろの基本アプリケーションをインストールします。

```
(chroot) livedd / # emerge --ask app-admin/syslog-ng
(chroot) livedd / # emerge --ask sys-process/cronie
(chroot) livedd / # emerge --ask sys-apps/mlocate net-misc/dhcpd
```

起動時に動くよう設定します。sshもついでに動くようにしましょう。

```
(chroot) livedd / # rc-update add syslog-ng default
* service syslog-ng added to runlevel default
(chroot) livedd / # rc-update add cronie default
* service cronie added to runlevel default
(chroot) livedd / # rc-update add sshd default
* service sshd added to runlevel default
```

ブートローダーの設定です。grubをインストールし、sdaにブートローダーをインストール。ブートするカーネルを設定しています。

```
(chroot) livedd / # emerge --ask sys-boot/grub
(chroot) livedd / # grub2-install /dev/sda
Installing for i386-pc platform.
Installation finished. No error reported.
(chroot) livedd / # grub2-mkconfig -o /boot/grub/grub.cfg
Generating grub configuration file ...
Linux イメージを見つけました: /boot/vmlinuz-4.1.12-gentoo
完了
```

8. 後片付け

これでひと通りの設定が完了です。まず今の環境から抜けます。

```
(chroot) livedd / # exit
```

マウントしていたファイルシステムをアンマウントします。

```
# umount -l /mnt/gentoo/{dev,sys}
# umount /mnt/gentoo/{proc,boot,gentoo}
```

最後にrebootします。次はインストールした環境で起動するはずです。

9. その後の設定

無事起動しましたか？もし起動しなかった場合、もう一度ブートCDで起動して環境を見なおしてみてください。再びchrootすれば途中からやり直せるはずです。それでもうまく行かない場合、私(@Wizard_of_Oz_)までメッセージを投げてください。手元で試しながら記載しましたが、もしかしたら間違っているところがあるかもしれませんので。

さて、ここままで基本のシステムはインストールできたのですが、まだまだ設定しなければいけないことがあります。ここからは時間の関係上(現在開催日当日午前3時です・・・)コマンドに一言添える程度の解説になります。不明な点あれば(山ほどあると思いますが)質問していただければ出来る限り答えますので、まずは参考情報として見てみてください。

9.1. rootアクセス禁止

インストール環境にsshしてつながらない場合、昔の接続情報が残っているのが原因の場合がある。下記のようにして接続情報を削除する。(接続先がハックされた場合なども同様の現象が起きる場合があります、確実に信頼できる環境に対して実施してください)

```
$ ssh-keygen -R 192.168.0.1(IPまたはホスト名)
ssh-keygen -R 192.168.0.1
# Host 192.168.0.1 found: line 11
/home/gentoo/.ssh/known_hosts updated.
Original contents retained as /home/gentoo/.ssh/known_hosts.old
```

9.2. SWAPファイル作成

```
gentoo / # fallocate -l 512M /swapfile
gentoo / # ls -lh swapfile
-rw-r--r-- 1 root root 512M 12月 30 01:43 swapfile
gentoo / # chmod 600 swapfile
gentoo / # ls -l swapfile -h
-rw----- 1 root root 512M 12月 30 01:43 swapfile
gentoo / # mkswap /swapfile
```

```
スワップ空間バージョン 1 を設定します。サイズ = 512 MiB (536866816 バイト)
ラベルはありません, UUID=d6407adb-333f-4180-ad9a-24f4bb6986aa
gentoo / # swapon /swapfile
gentoo / # swapon
NAME      TYPE  SIZE USED PRIO
/swapfile file 512M  0B   -1
gentoo / # nano /etc/fstab
gentoo / # cat /etc/fstab
# NOTE: If your BOOT partition is ReiserFS, add the notail option to opts.
/dev/sda2          /boot          ext2          noauto,noatime 1 2
/dev/sda3          /              ext4          noatime        0 1
#/dev/SWAP         none           swap          sw             0 0
/swapfile          none           swap          defaults       0 0
/dev/cdrom         /mnt/cdrom     auto          noauto,ro      0 0
#/dev/fd0          /mnt/floppy    auto          noauto         0 0
```

9.3. NTP設定

```
# emerge -av net-misc/ntp
# ntpdate ntp.nict.jp

# nano /etc/conf.d/ntp-client
NTPCLIENT_OPTS='-s -b -u ntp.nict.jp'

# rc-update add ntp-client default
* service ntp-client added to runlevel default
```

9.4. CPUフラグの設定

使用しているCPUにあったフラグを/etc/portage/make.confファイルに設定する。

```
CPU_FLAGS_X86="mmx mmxext sse sse2 sse3"
```

現状ではUSEにも同じものを設定しておいたほうがいい。

```
USE="mmx mmxext sse sse2 sse3"
```

設定するフラグはcat /proc/cpuinfoでも確認できるが、自動的に設定値を調べてくれるcpuinfo2cpuflags-x86というプログラムがあるのでこれを使用するのが楽。

```
emerge -1v app-portage/cpuinfo2cpuflags
cpuinfo2cpuflags-x86 >> make.conf
```

9.5. PortageのSYNC設定

Portageは日々更新されるため、アップデートする必要がある。現在はrepos.confディレクトリにSync情報を記載する。デフォルトの設定ファイルがあるので以下のようにコピーする。

```
gentoo ~ # mkdir /etc/portage/repos.conf
gentoo ~ # cp /usr/share/portage/config/repos.conf /etc/portage/repos.conf/
gentoo.conf
gentoo ~ # cat /etc/portage/repos.conf/gentoo.conf
[DEFAULT]
main-repo = gentoo

[gentoo]
location = /usr/portage
sync-type = rsync
sync-uri = rsync://rsync.gentoo.org/gentoo-portage
auto-sync = yes

# for daily squashfs snapshots
#sync-type = squashdelta
#sync-uri = mirror://gentoo/./snapshots/squashfs
```

rsync

今までどおりrsyncを使用する場合、設定ファイルのsync-uriを日本のものに設定する。

```
sync-uri = rsync://rsync.gentoo.org/gentoo-portage
```

```
sync-uri = rsync://rsync.jp.gentoo.org/gentoo-portage
```

下記コマンドでsyncされる

```
gentoo ~ # emerge --sync
```

git

SYNCにgitを使用する場合、下記手順を行う。gitのほうがrsyncよりも数倍早い。

```
gentoo ~ # emerge -av dev-vcs/git
```

gentoo.confを修正。

```
sync-type = git
sync-uri = https://github.com/gentoo-mirror/gentoo
```

このままSYNCすると怒られる。

```
gentoo ~ # emerge --sync
>>> Syncing repository 'gentoo' into '/usr/portage'...
/usr/bin/git clone --depth 1 https://github.com/gentoo-mirror/gentoo .
fatal: destination path '.' already exists and is not an empty directory.
!!! git clone error in /usr/portage
```

一旦portageを移動する。うまく言ったら古い方は消して良い。

```
gentoo ~ # mv /usr/portage{,.old}

gentoo ~ # emerge --sync
>>> Syncing repository 'gentoo' into '/usr/portage'...
/usr/bin/git clone --depth 1 https://github.com/gentoo-mirror/gentoo .
Cloning into '.'...
remote: Counting objects: 155447, done.
remote: Compressing objects: 100% (127329/127329), done.
remote: Total 155447 (delta 29277), reused 114064 (delta 26893), pack-reused 0
Receiving objects: 100% (155447/155447), 79.52 MiB | 8.38 MiB/s, done.
Resolving deltas: 100% (29277/29277), done.
Checking connectivity... done.
Checking out files: 100% (142225/142225), done.
=== Sync completed for gentoo
```

```
Performing Global Updates
(Could take a couple of minutes if you have a lot of binary packages.)
.='update pass' *='binary update' #='/var/db update' @='/var/db move'
s='/var/db SLOT move' %='binary move' S='binary SLOT move'
p='update /etc/portage/package.*'
/usr/portage/profiles/
updates/1Q-2010.....
/usr/portage/profiles/updates/4Q-2015.....
```

9.6. プログラム検索ツールeix

emerge -S <name>でプログラムを検索できるが、非常に遅い。 eixを使用すると非常に高速に検索ができる。


```
# emerge -av app-portage/eix
```

インストール直後はキャッシュが作成されていないため怒られる。

```
gentoo ~ # eix emacs
Cannot open the database file /var/cache/eix/portage.eix for reading.
Did you forget to create it with 'eix-update'?
```

指示の通りeix-updateを実行しても良いが、eix-syncを使用すると emerge --syncとeix-updateを療法行ってくれるので、毎回これを実行しておけばよい。

```
gentoo ~ # eix-sync
* eix-cache does not exist.
* Running eix-update
Reading Portage settings ..
Building database (/var/cache/eix/portage.eix) ..
[0] 'gentoo' /usr/portage/ (cache: metadata-md5-or-flat)
Reading category 163|163 (100%) Finished
Applying masks ..
Calculating hash tables ..
Writing database file /var/cache/eix/portage.eix ..
Database contains 18789 packages in 163 categories.
* Running emerge --sync
>>> Syncing repository 'gentoo' into '/usr/portage'...
/usr/bin/git pull
Already up-to-date.
=== Sync completed for gentoo

* Main gentoo tree does not appear to have changed: exiting
* Use -a or set have_changed=: in a ! hook to override this check
* Time statistics:
16 seconds for initial eix-update
2 seconds for syncing
19 seconds total
```

9.7. VMwareでの便利なツール(VMware-tools)

VMware-toolsにはVMware Playerに付いているもの、(おそらく)それをemergeでインストールできるようにしたもの、オープンソースのものなど複数あるが、オープンソースのものを利用するのが主流。

```
gentoo ~ # eix open-vm-tools
```

```
[I] app-emulation/open-vm-tools
Available versions: ~9.4.0.1280544 ~9.10.0_p2476743 ~9.10.2_p2822639
(~)10.0.0_p3000743 {X doc fuse grabbitmqproxy icu modules pam +pic vgauth
xinerama}
Homepage:          https://github.com/vmware/open-vm-tools
Description:       Opensourced tools for VMware guests

* app-emulation/open-vm-tools-kmod
Available versions: ~9.4.0.1280544 ~9.10.0_p2476743 ~10.0.0_p3000743 {+vmhgfs
+vmxnet KERNEL='linux'}
Homepage:          http://open-vm-tools.sourceforge.net/
Description:       Opensourced tools for VMware guests

Found 2 matches.
```

open-vm-toolsのみインストール。

```
gentoo ~ # emerge -av open-vm-tools

* IMPORTANT: 9 news items need reading for repository 'gentoo'.
* Use eselect news read to view new items.

These are the packages that would be merged, in order:

Calculating dependencies... done!
[ebuild N    ] sys-apps/ethtool-3.18::gentoo 195 KiB
[ebuild N    ] dev-libs/libdnet-1.12::gentoo USE='ipv6 -python -static-libs {-
test}' PYTHON_TARGETS='python2_7' 953 KiB
[ebuild N    ] sys-fs/fuse-2.9.4::gentoo USE='-examples -static-libs' 564 KiB
[ebuild N    ~] app-emulation/open-vm-tools-10.0.0_p3000743::gentoo USE='pam pic
-X -doc -grabbitmqproxy -icu -vgauth -xinerama' 4,064 KiB

Total: 4 packages (4 new), Size of downloads: 5,774 KiB

The following keyword changes are necessary to proceed:
(see ``package.accept_keywords'' in the portage(5) man page for more details)
# required by open-vm-tools (argument)
=app-emulation/open-vm-tools-10.0.0_p3000743 ~amd64

would you like to add these changes to your config files? [Yes/No]
```

open-vm-toolsはテストが十分ではないので、~amd64オプションをつけると言われている。Yを選択すると自動的に設定(の準備)がされる。

```
would you like to add these changes to your config files? [Yes/No] y
```

```
Autounmask changes successfully written.
```

```
* IMPORTANT: config file '/etc/portage/package.accept_keywords' needs updating.  
* See the CONFIGURATION FILES section of the emerge  
* man page to learn how to update config files.
```

実際にはまだ反映されておらず、dispatch-confコマンドで反映される。対話形式でどのように設定ファイルを修正するか聞いてくれる。最初は何も考えずにuを連打しておけばいい。

```
gentoo ~ # dispatch-conf  
  
--- /tmp/tmpxkh79313/0 2015-12-30 20:16:01.200059604 +0900  
+++ /etc/portage/.cfg0000_package.accept_keywords 2015-12-30  
20:15:34.240059185 +0900  
@@ -1 +1,2 @@  
-/dev/null  
+# required by open-vm-tools (argument)  
+=app-emulation/open-vm-tools-10.0.0_p3000743 ~amd64  
  
>> (1 of 1) -- /etc/portage/package.accept_keywords  
>> q quit, h help, n next, e edit-new, z zap-new, u use-new  
m merge, t toggle-merge, l look-merge:
```

再度インストールを実行するとインストールできるようになっている。

```
gentoo ~ # emerge -av open-vm-tools
```

open-vm-toolsをインストールすると以下のことができるようになる。

ホストとのファイル共有

仮想マシン設定→オプション にある共有フォルダが使用できるようになる。共有フォルダを設定しておき、vmware-hgfsclientコマンドを実行すると 設定されている共有フォルダの情報が見える。

```
# vmware-hgfsclient  
share
```

下記コマンドでマウントできる。

```
vmhgfs-fuse .host:/share mnt
```

vmware-toolbox-cmd

いろいろな便利ツール

```
# vmware-toolbox-cmd help
```

でヘルプが見える。 便利なのが以下のコマンド。仮想ディスクは一旦ファイルを書込サイズが大きくなると ファイルを削除しても縮まらない。下記コマンドを使用すると縮めてくれる。

```
# vmware-toolbox-cmd disk shrinkonly
```

その他

ドラッグアンドドロップを使用できるようにしてくれる模様。調査中。

```
# vmware-vmblock-fuse  
# service vmware-tools start
```

open-vm-tools-kmodはドライバの模様。インストールすると成功はしたが下記メッセージが出た。 ちゃんと使うにはカーネルコンフィグして下記を有効にする必要がある。BALLOONとか何が変わるのか気になる・・・

```
* Messages for package app-emulation/open-vm-tools-kmod-10.0.0_p3000743:  
  
* CONFIG_DRM_VMWGFX: is not set when it should be.  
* CONFIG_VMWARE_BALLOON: is not set when it should be.  
* CONFIG_VMWARE_PVSCSI: is not set when it should be.  
* CONFIG_VMXNET3: is not set when it should be.  
* CONFIG_VMWARE_VMCI: is not set when it should be.  
* CONFIG_VMWARE_VMCI_VSOCKETS: is not set when it should be.  
* Please check to make sure these options are set correctly.  
* Failure to do so may cause unexpected problems.
```

9.8. GUI(X)の設定

Xfceのインストールを例に上げる。 <https://wiki.gentoo.org/wiki/Xfce/ja> を見ながらやりました。

まずはプロファイルの変更。

```
gentoo ~ # eselect profile list
Available profile symlink targets:
[1]  default/linux/amd64/13.0 *
[2]  default/linux/amd64/13.0/selinux
[3]  default/linux/amd64/13.0/desktop
[4]  default/linux/amd64/13.0/desktop/gnome
[5]  default/linux/amd64/13.0/desktop/gnome/systemd
[6]  default/linux/amd64/13.0/desktop/kde
[7]  default/linux/amd64/13.0/desktop/kde/systemd
[8]  default/linux/amd64/13.0/desktop/plasma
[9]  default/linux/amd64/13.0/desktop/plasma/systemd
[10] default/linux/amd64/13.0/developer
[11] default/linux/amd64/13.0/no-multilib
[12] default/linux/amd64/13.0/systemd
[13] default/linux/amd64/13.0/x32
[14] hardened/linux/amd64
[15] hardened/linux/amd64/selinux
[16] hardened/linux/amd64/no-multilib
[17] hardened/linux/amd64/no-multilib/selinux
[18] hardened/linux/amd64/x32
[19] hardened/linux/musl/amd64
[20] hardened/linux/musl/amd64/x32
[21] default/linux/uclibc/amd64
[22] hardened/linux/uclibc/amd64
gentoo ~ # eselect profile set 3
```

make.confのUSEにXを追加、VIDEO_CARDS="vmware"の行を追加

```
gentoo ~ # nano /etc/portage/make.conf
gentoo ~ # cat /etc/portage/make.conf
# These settings were set by the catalyst build script that automatically
# built this stage.
# Please consult /usr/share/portage/config/make.conf.example for a more
# detailed example.
CFLAGS='-O2 -pipe'
CXXFLAGS='${CFLAGS}'
MAKEOPTS='-j4'
# WARNING: Changing your CHOST is not something that should be done lightly.
# Please consult http://www.gentoo.org/doc/en/change-chost.xml before changing.
CHOST='x86_64-pc-linux-gnu'
# These are the USE flags that were used in addition to what is provided by the
# profile used for building.
USE='bindist mmx sse sse2 X'
PORTDIR='/usr/portage'
DISTDIR='${PORTDIR}/distfiles'
PKGDIR='${PORTDIR}/packages'
```

```
VIDEO_CARDS='vmware'
```

xorg-x11をインストール

```
# emerge -av xorg-x11
```

下記はインストールログ メモとして残す

```
* Messages for package media-fonts/liberation-fonts-2.00.1-r1:

* The following fontconfig configuration files have been installed:
*
*   60-liberation.conf
*
* Use `eselect fontconfig` to enable/disable them.

* Messages for package x11-libs/libxi-1.7.5:

* Some special keys and keyboard layouts may stop working.
* To fix them, recompile xorg-server.

* Messages for package media-libs/mesa-11.0.6:

* Note that in order to have full S3TC support, it is necessary to install
* media-libs/libtxc_dxtn as well. This may be necessary to get nice
* textures in some apps, and some others even require this to run.

* Messages for package x11-apps/xinit-1.3.4-r1:

* If you use startx to start X instead of a login manager like gdm/kdm,
* you can set the XSESSION variable to anything in /etc/X11/Sessions/ or
* any executable. When you run startx, it will run this as the login session.
* You can set this in a file in /etc/env.d/ for the entire system,
* or set it per-user in ~/.bash_profile (or similar for other shells).
* Here's an example of setting it for the whole system:
*   echo XSESSION='Gnome' > /etc/env.d/90xsession
*   env-update && source /etc/profile

* Messages for package media-libs/fontconfig-2.11.1-r2:

* Please make fontconfig configuration changes using `eselect
* fontconfig`. Any changes made to /etc/fonts/fonts.conf will be
* overwritten. If you need to reset your configuration to upstream
* defaults, delete the directory /etc/fonts/conf.d/ and re-emerge
* fontconfig.
*
* (Note: Above message is only printed the first time package is
```

```
* installed. Please look at /usr/share/doc/fontconfig-2.11.1-r2/README.gentoo*
* for future reference)

* Messages for package x11-base/xorg-x11-7.4-r2:

*
* Please note that the xcursors are in /usr/share/cursors/xorg-x11.
* Any custom cursor sets should be placed in that directory.
*
* If you wish to set system-wide default cursors, please create
* /usr/local/share/cursors/xorg-x11/default/index.theme
* with content: ``Inherits=theme_name'' so that future
* emerges will not overwrite those settings.
*
* Listening on TCP is disabled by default with startx.
* To enable it, edit /usr/bin/startx.
*
* Visit https://www.gentoo.org/doc/en/index.xml?catid=desktop
* for more information on configuring X.
*
```

qt4,qt5を無効化

```
nano /etc/portage/package.use/xfce
app-text/poppler -qt4 -qt5
dev-util/cmake -qt4 -qt5
```

make.confに下記行を追加

```
XFCE_PLUGINS=''brightness clock trash''
```

インストール

```
gentoo portage # emerge -av xfce-meta xfce-extra/xfce4-notifyd
gentoo portage # emerge -av --deselect=y xfce-extra/xfce4-notifyd
>>> Removing xfce-extra/xfce4-notifyd from ``world'' favorites file...
Would you like to remove these packages from your world favorites? [Yes/No] y
```

これでVM上でstartxfce4コマンドを実行するとGUIが起動するようになる。

とりあえず端末を入れる

```
gentoo ~ # emerge -av xfce4-terminal
```

open-vm-toolsを入れなおす。Xに関する機能が入る。これでマウスがVM-ホストをシームレスに動けるようになる(少し引かかる所があるけど)。ドラッグアンドドロップ、画面サイズの可変も可能になっていた。(要調査)

```
gentoo ~ # emerge -av open-vm-tools
```

もし画面サイズの可変が動いていなかった場合、以下のコマンドで使えるようになる。

```
$ vmware-user-suid-wrapper
```

9.9. 日本語入力

```
# emerge -av ibus mozc
# nano ~/.xprofile (or .xinitrc)
export XMODIFIERS="@im=ibus"
export GTK_IM_MODULE="ibus"
export Qt_IM_MODULE="xim"
ibus-daemon -d -x
```

9.10. ブラウザ

chromiumやfirefoxはかなり大きいソフトウェアなので、コンパイルに非常に時間がかかる。Gentooは一部のアプリにバイナリも提供しているので、すぐ利用したい場合は以下を利用してみよう。

- google-chrome
- firefox-bin

10. あとがき

現在12月31日午前6時です。ここまで見ていただいた皆さん申し訳ありません。
mikutterインストール完了まで書ききれませんでした・・・

口調が定まっていなかったり、途中から解説ないまま超高速で駆け抜けてしまっていますが、なにか少しでも役に立つ部分があるといいのですが。元々の予定では、各種インストール済みのVMもともに配布予定でしたが、このざまなので作っている時間がありませんでした。もしも欲しいという方がいれば、twitterで連絡ください。

今回インストール手順書を書くため、自分の中でまとまっていなかった知識を整理すると詳しく分からなかったり新たな発見があったりして調べる方に夢中になり、現行にとりかかったのが本番数日前というありさまでした。とくにVMwareツールはあまり情報がなく、今回多少なりともまとめたのは良かったかなと思います。

このようにGentooは調べれば調べるほど深みにはまっていく楽しさがあるので、ぜひともみなさんこの沼に沈んでみませんか(^o^)

さて、上記すべての作業が(不足した情報を埋めて)実施できたとすると、GentooInstallBattleの残りの作業はmikutterそのもののインストールのみとなるはずです。最後に以下のコマンドを実行してみましょう。

```
# emerge -av net-misc/mikutter
```

open-vm-toolsのところで説明したように、amd64オプション追加が(大量に)出るかと思います。yesを押してdispatch-confを実行しuを押して全て反映させましたか？もう一度コマンドを実行してみてください。インストール成功しましたか？失敗しましたか？インストール成功した場合、mikutterを起動し、アカウントを登録してツイートしてください。これにて#GentooInstallBattleは終了です！！お疲れ様でした！！

もし失敗した場合、此处から先はあなたが調査してみてください(そして結果を教えてください:-p) もちろん調べて分からなかった場合、私の方まで連絡いただければこちらでも調べてみます。此处から先はGentoo Wiki(やarch Wiki)を見て好きな環境を構築してみてください。そしてその情報をWebやTwitterにでも発信して頂けるとありがたいです。

11. emergeチートシート

ページが余ったので

■SYNC

Portageツリーの更新(`apt-get update`相当)

```
# emerge --sync
```

```
# eix-sync
```

■調査

ソフトウェアの検索

```
# emerge --search(-s) <名前(の一部)>
```

```
# eix <名前(の一部)>
```

インストールするパッケージの確認

```
# emerge --verbose(-v) --pretend(-p) <package name>
```

■インストール

パッケージのインストール

```
# emerge <package name>
```

インストールされるものを確認してからインストール

```
# emerge --ask -v <package name>
```

インストールするものをworldファイルに書き込まない(明示的インストールにしない)
依存関係で入れるものを(emerge失敗などで)手動で入れるときなど

```
# emerge --oneshot(-1) <package name>
```

ビルド失敗しても最後まで止めないオプション

GUIやアップデートなど大量のインストールがあるときに、 朝起きたらすぐ止まって泣かないように

```
# emerge --keep-going
```

インストール除外するオプション

@worldでアップデートすると困るものに

```
# emerge --exclude=<package name>
```

■アンインストール

指定したパッケージを削除する(依存関係は削除されない)

```
# emerge --unmerge(-C) <package name>
```

不要になったパッケージを削除する

unmerge後などに実行

Kernelも最新以外削除されるので注意

```
# emerge --depclean -a
```

■アップデート

指定したパッケージをアップデート

```
# emerge --update(-u) <package name>
```

自分がインストールしたパッケージをアップデート(依存で入ったものはアップデートされない)

```
# emerge -au @world
```

インストールしたすべてのパッケージをアップデート(依存で入ったものもアップデートされる)

```
# emerge -au --deep(-D) @world
```

USEフラグに変更があった場合もアップデート

```
# emerge -auD --newuse(-N) @world
```

略記号からうどんあつぷでーなどと呼ばれる
時々これを実行しておけば最新を追従できる

```
# emerge -auDN @world
```

依存関係をさかのぼってアップデート

依存関係の問題でうまくアップデートができないとき試すとうまくいくことが多い

```
# emerge --emptytree(-e) <package name>
```

すべてアップデート(というか再インストール)

部屋があったかくなる

```
# emerge -e @world
```