

インストール

- ・ デスクトップまたはサーバ (GUI) としてインストール
- ・ kdump は無効

サーバの設定

sudo の設定

```
su  
visudo
```

パスワードを求める場合は以下をコメントイン

```
%wheel    ALL=(ALL)    ALL
```

パスワードを求めない場合は以下をコメントイン

```
%wheel    ALL=(ALL)    NOPASSWD: ALL
```

sudo の実行時は path が変わるので

```
secure_path
```

の行を探して、編集する。

backup

```
sudo vi /usr/local/bin/backup  
  
#!/bin/sh  
if [ -e "$1" ]; then  
    dir=.  
    if [ "$2" != "" ]; then  
        dir=$2  
    fi  
    op=  
    if [ -d "$1" ]; then  
        op=-r  
    fi  
  
    tofile=$dir/$1_$(date +%Y%m%d_%H%M%S)  
    cp -p $op $1 $tofile  
    echo "backup to $tofile"  
fi
```

yum の設定

media の追加

```
sudo vi /etc/yum.repos.d/CentOS-Media.repo
```

```
[media]
```

```
name=CentOS-$releasever - Media
baseurl=file:///media/CentOS/
        file:///media/cdrom/
        file:///run/media/centos/CentOS%207%20x86_64/
        file:///media/cdrecorder/
enabled=0
gpgcheck=0
# gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
```

epel の利用

```
sudo yum install epel-release
```

nux-dextop の利用

```
sudo rpm -Uvh http://li.nux.ro/download/nux/dextop/el7/x86_64/nux-dextop-release-0-5.el7
.nux.noarch.rpm
```

他のリポジトリ

<http://everything-you-do-is-practice.blogspot.com/2017/09/centos-epel-ius-remi-centos-5x-centos.html>

yum-cron

http://pentan.info/server/linux/yum_updatesd.html

<https://www.hiroom2.com/2016/05/27/centos-7-%E8%87%AA%E5%8B%95%E3%82%A2%E3%83%83%E3%83%97%E3%83%87%E3%83%BC%E3%83%88%E3%81%AE%E8%A8%AD%E5%AE%9A%E3%82%92%E3%81%99%E3%82%8B/>

とりあえず、自動更新はしない方向で運用。

なので、yum-cron も yum-updatesd もインストールしない。

グラフィックドライバ

オンボードのインテルチップを使う。

CentOS インストール時にドライバもインストールされるはず。

```
yum install xorg-x11-drv-intel
```

SELinux を無効にする

/etc/sysconfig/selinux

の

```
SELINUX=enforcing
```

を

```
SELINUX=disabled
```

に変更。

ホスト名

```
hostnamectl set-hostname ホスト名
```

確認

```
hostnamectl
```

```
/etc/hostname
```

を直接書きなおしても良い。上記コマンドでこのファイルも編集される。

ネットワーク関連

ネットワーク設定

NetworkManager ではマルチセッションが張れないなどの問題があるので、無効化して従来の network で設定する。

NetworkManager を無効にする

仮想ブリッジ virbr0 の無効化

仮想ブリッジ virbr0 の無効化と有効化

ファイアウォールの設定

firewalld を止めて、iptables コマンドを使う。

毎回起動時に iptables コマンドで設定するので、iptables-services は不要だけども一応入れておく。

```
yum install iptables-services
systemctl disable firewalld
systemctl enable iptables-services
```

iptables でファイヤーウォール

SSH

SSH サーバーの設定

DNS

DNS サーバーの設定

DHCP サーバーの設定

DHCP サーバーの設定

samba サーバーの設定

samba サーバーの設定

sendmail の設定

サービス等からメールを送信するために smtp として gmail を使う。

sendmail コマンドの smtp サーバの設定

ハードディスク監視

HDD のスピンドアウン設定

Linux でアクセスの無い HDD をスピンドアウンする

smartd の設定

Linux で HDD の SMART をモニタリングする

locate の検索対象

/mnt などが検索対象外になっていることを確認する。

Linux で locate を使ってファイルを検索する

バックアップジョブ

システムバックアップ

システムバックアップ XFS

データバックアップ

ディレクトリをバックアップする

crontab でスケジューリング

```
crontab -e
```

夜中 3 時にバックアップ開始

```
00 03 * * * /mnt/sdb/backup/backup.sh
```

または、

```
vim /etc/crontab  
00 03 * * * root /mnt/sdb/backup/backup.sh
```

時刻同期

CentOS7 では ntpd ではなく、chrony に変わった。

ntpd は不要。

```
systemctl status chronyd  
systemctl enable ntpd
```

デスクトップの設定

bash の設定

.bashrc

```
# .bashrc  
  
# Source global definitions  
if [ -f /etc/bashrc ]; then  
    . /etc/bashrc  
fi  
  
# 新しく作られたファイルのパーミッションがつねに 644 になるようにする  
umask 022  
# core ファイルを作らせないようにする  
ulimit -c 0  
  
# export GTK_IM_MODULE=ibus
```

```

# export XMODIFIERS=@im=ibus
# export QT_IM_MODULE=ibus

# 非対話モードの場合は終了
if [[ ! "${PS1}" ]]; then
    return
fi

#####
# 環境ごとの設定
#####
if [[ "${MSYSTEM}" ]]; then
    export COPY_CLIP='eval nkf -s | clip'
    alias peco='percol'
    GIT_PROMPT=/usr/share/git/completion/git-prompt.sh
    export DISPLAY=localhost:0.0
elif [[ "$(uname)" == 'Darwin' ]]; then
    export COPY_CLIP='pbcopy'
    GIT_PROMPT=/usr/local/etc/bash_completion.d/git-prompt.sh
    if [ -f $(brew --prefix)/etc/bash_completion ]; then
        . $(brew --prefix)/etc/bash_completion
    fi
else
# elif [[ "$(expr substr $(uname -s) 1 5)" == 'Linux' ]]; then
if [ -f /mnt/c/Windows/system32/clip.exe ]; then
    export COPY_CLIP='clip.exe'
else
    export COPY_CLIP='xclip -i -sel clip'
fi
GIT_PROMPT=/usr/share/git-core/contrib/completion/git-prompt.sh
fi

#####
# 基本的な設定
#####
export PAGER='less -m -i'

# EOF (Ctrl-D) 入力は 10 回まで許可。
IGNOREEOF=10
# 履歴のサイズ
HISTSIZE=1000
# 履歴ファイルを上書きではなく追加する。
shopt -s histappend
# 実行するまえに必ず展開結果を確認できるようにする。
shopt -s histverify
# 履歴の置換に失敗したときやり直せるようにする。
shopt -s histreedit
# 履歴の重複を削除
export HISTCONTROL=erasedups      #ignoredups,ignorespace,erasedups

bind '"%C-n": history-search-forward'
bind '"%C-p": history-search-backward'

#####
# alias の設定
#####
alias less='less -i'
alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'
# alias for misc
alias grep='grep --color'
# alias for some shortcuts for different directory listings
if [[ "$(uname)" == 'Darwin' ]]; then
    export LSCOLORS=gxfxcxdxbxegedabagacad
    alias ls='ls -G'
else
    export LS_COLORS='di=01;36:ow=30;42'
    alias ls='ls -hF --color=auto --show-control-chars'
fi
alias ll='ls -l'
alias la='ls -A'
alias l='ls -CF'
alias tig='tig --date-order'
# alias cdd='cd $(find . -maxdepth 1 -type d | peco)'
alias cdd='cd'
_cdd() {
    _cd
    COMPREPLY=($(IFS=$'\n'; echo "${COMPREPLY[*]}") | peco)
}
complete -F _cdd cdd

```

```
#####
# peco で履歴をインクリメントサーチ
#####
peco-select-history() {
    declare l=$(HISTTIMEFORMAT= history | sort -k1,1nr | perl -ne 'BEGIN { my @lines = (); }
s/^s*d+s*//; $in=$_; if (!(grep {$in eq $_} @lines)) { push(@lines, $in); print $in; }' | peco
--query "$READLINE_LINE")
    READLINE_LINE="$l"
    READLINE_POINT=${#l}
}
bind -x '"C-r": peco-select-history'

#####
# git
#####
gitps1=true
if [[ -r ${GIT_PROMPT} ]]; then
    . ${GIT_PROMPT}
    gitps1="__git_ps1"
fi
GIT_PS1_SHOWDIRTYSTATE=true
GIT_PS1_SHOWUNTRACKEDFILES=true
GIT_PS1_SHOWSTASHSTATE=true
GIT_PS1_SHOWUPSTREAM=auto

if [[ "${MSYSTEM}" ]]; then
    export PS1='${0};$TITLEPREFIX:$PWD%007%$n%[%03[32m%]$u%h %[%03[35m%]$MSYSTEM %[%03[33
m%]$w%[%03[36m%]' ${gitps1} %[%03[0m%]$n$ '
else
    export PS1='${u%h %[%03[33m%]$W%[%03[36m%]$( ${gitps1} )%[%03[0m%]}'$ '
fi
# デフォルト
# '${u%h $W}%$ '

#####
# ssh agent
#####
agent="$HOME/.ssh/agent"
if [ -S "$SSH_AUTH_SOCK" ]; then
    case $SSH_AUTH_SOCK in
        /tmp/*agent.[0-9]*)
            ln -snf "$SSH_AUTH_SOCK" $agent && export SSH_AUTH_SOCK=$agent
    esac
elif [ -S $agent ]; then
    export SSH_AUTH_SOCK=$agent
fi

if [[ ! "${TMUX}" ]] && type -a tmux > /dev/null 2>&1; then
    tmux new-session
fi
```

.bashrc(以前使っていたもの)

<http://www.unixuser.org/~euske/doc/bashtips/bashrc.html>

```
## 新しく作られたファイルのパーミッションがつねに 644 になるようにする。
umask 022
## core ファイルを作らせないようにする。
ulimit -c 0

export PAGER='less -m -i'
export LS_COLORS='di=01;36'

if [[ "${PS1}" ]]; then
    # EOF (Ctrl-D) 入力は 10 回まで許可。
    IGNOREEOF=10
    # 履歴のサイズ
    HISTSIZE=100
    # 履歴ファイルを上書きではなく追加する。
    shopt -s histappend
    # 実行するまえに必ず展開結果を確認できるようにする。
    shopt -s histverify
    # 履歴の置換に失敗したときやり直せるようにする。
    shopt -s histreedit
    # 履歴の重複を削除
    export HISTCONTROL=erasedups    #ignoredups,ignorespace,erasedups

    bind '"C-n": history-search-forward'
```

```

bind '"␣-p": history-search-backward'

# alias
# for interactive operation
alias less='less -i'
alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'
# alias for misc
alias grep='grep --color'
# alias for some shortcuts for different directory listings
alias ls='ls -hF --color=always --show-control-chars'
alias ll='ls -l'
alias la='ls -A'
alias l='ls -CF'
# alias peco=percol
# alias cdd='cd $(find . -maxdepth 1 -type d | peco)'
fi

# peco で履歴をインクリメントサーチ
# peco-select-history() {
#     declare l=$(HISTTIMEFORMAT= history | sort -k1,1nr | perl -ne 'BEGIN { my @lines = (); }
# s/^s*d+s*//; $in=$_; if (!(grep {$in eq $_} @lines)) { push(@lines, $in); print $in; }' | peco
--query "$READLINE_LINE")
#     READLINE_LINE="$l"
#     READLINE_POINT=${#l}
# }
# bind -x '"␣-r": peco-select-history'

# git
gitps1=true
GIT_CONTRIB_DIR=/usr/share/git-core/contrib
if [ -r ${GIT_CONTRIB_DIR}/completion/git-prompt.sh ]; then
    . ${GIT_CONTRIB_DIR}/completion/git-prompt.sh
gitps1="__git_ps1"
fi
GIT_PS1_SHOWDIRTYSTATE=true
GIT_PS1_SHOWUNTRACKEDFILES=true
GIT_PS1_SHOWSTASHSTATE=true
GIT_PS1_SHOWUPSTREAM=auto

export PS1='[%u@%h %W$( ${gitps1} )]%'
# デフォルト
# '[%u@%h %W]%'

```

.inputrc

inputrc

~/ssh/config

SSH クライアントの設定

ウィンドウマネージャの設定

ログイン画面

もし、ログイン画面でユーザ名も入力させる（ユーザ名を選ばせない）場合は

```
sudo -u gdm dbus-launch gsettings set org.gnome.login-screen disable-user-list true
```

を実行する。

文字が小さい場合は、ログイン画面右上のアクセシビリティアイコンを選択して

大きな文字

を選択する。倍率は 1.2 倍固定だが以下のコマンドで倍率を変更できる。

ただし、大きな文字を選択し直すと 1.2 倍に戻ってしまう。

```
sudo -u gdm dbus-launch gsettings set org.gnome.desktop.interface text-scaling-factor '1.5'
```

GNOME3

CentOS7 で GNOME3 を使う

MATE

CentOS7 で MATE を使う

Xfce

CentOS7 で Xfce を使う

ワークスペース毎の壁紙変更

Gnome でワークスペース毎に壁紙を変更する

クリップボードマネージャ

Linux でクリップボード管理

tmux

tmux

XMMS

CentOS で xmms を使う

VLC

CentOS で vlc player を使う

プライバシーの設定

基本的にすべて無効

日本語入力

CentOS で Mozc を使う