

2.9

インストール

OS のクリップボードとの共有に xclip を使う

```
$ sudo yum install libevent-devel xclip ncurses-devel byacc
$ git clone https://github.com/tmux/tmux.git
$ cd tmux
$ sh autogen.sh
$ ./configure && make
```

~/.tmux.conf

```
# tmux list-keys

# 設定ファイルをリロードする
bind r source-file ~/.tmux.conf ¥; display "Reloaded!"

# 色が正常に表示されないことへの対応
set -g default-terminal "screen-256color"
set -g terminal-overrides 'xterm:colors=256'

# pane, window
bind -n M-Pageup select-pane -t :.-
bind -n M-PageDown select-pane -t :.+
bind -n C-Pageup select-window -t :.-
bind -n C-PageDown select-window -t :.+
bind -n M-p select-pane -t :.-
bind -n M-@ select-pane -t :.+
bind -n M-P select-window -t :.-
bind -n M-` select-window -t :.+

# ウィンドウ作成
bind c new-window -c '#{pane_current_path}'
bind-key m if-shell "tmux show-options -g mouse | grep off" "set-option -g mouse on ¥; display
¥"mouse on¥" "set-option -g mouse off ¥; display ¥"mouse off¥"

# Keybind ウィンド分割
bind 3 split-window -h -c '#{pane_current_path}'
bind | split-window -h -c '#{pane_current_path}'
bind 2 split-window -v -c '#{pane_current_path}'
bind - split-window -v -c '#{pane_current_path}'

# バッファ数 (デフォルト 2000 行)
# set-option -g history-limit 5000

# メッセージ表示時間
set-option -g display-time 10000
set-option -g display-panes-time 10000

# マウス操作 (コメントアウトすればターミナルのマウス操作ができるようになる)
set-option -g mouse on

# Window 名の自動設定 off
set-option -g allow-rename off

# ステータスバーをトップに配置する
# set-option -g status-position top
# set-option -g status-bg "colour51"

## ウィンドウのインデックスを 1 から始める
set -g base-index 1
## ペインのインデックスを 1 から始める
setw -g pane-base-index 1
## アクティブなペインのみ白っぽく変更 (真っ黒は 232)
set -g window-style 'bg=#212121'
set -g window-active-style 'bg=#000000'

# ステータスバー
set-option -g status-left '[#S]'
set-option -g status-right "[%Y-%m-%d %H:%M]"

# コピペ
bind v choose-buffer
```

```

set-window-option -g mode-keys vi
unbind -T copy-mode-vi MouseDragEnd1Pane
unbind-key -T copy-mode-vi Home
bind-key -T copy-mode-vi Home send -X start-of-line
unbind-key -T copy-mode-vi End
bind-key -T copy-mode-vi End send -X end-of-line
unbind-key -T copy-mode-vi C-Left
bind-key -T copy-mode-vi C-Left send -X previous-word
unbind-key -T copy-mode-vi C-Right
bind-key -T copy-mode-vi C-Right send -X next-word
unbind-key -T copy-mode-vi C-Home
bind-key -T copy-mode-vi C-Home send -X history-top
unbind-key -T copy-mode-vi C-End
bind-key -T copy-mode-vi C-End send -X history-bottom
# bind-key -T copy-mode-vi Enter send-keys -X copy-pipe "tmux copy-buffer"
bind-key -T copy-mode-vi v send-keys -X begin-selection
bind-key -T copy-mode-vi y send-keys -X copy-pipe-and-cancel "${COPY_CLIP}"
bind-key -T copy-mode-vi Enter send-keys -X copy-pipe-and-cancel "${COPY_CLIP}"
bind-key -T copy-mode-vi c send-keys -X copy-pipe "${COPY_CLIP}" ¥; display "Copy"

bind-key -T copy-mode-vi C-v send -X rectangle-toggle

# List of plugins
set -g @plugin 'tmux-plugins/tpm'
set -g @plugin 'tmux-plugins/tmux-sensible'
set -g @plugin 'tmux-plugins/tmux-copycat'

# Initialize TMUX plugin manager (keep this line at the very bottom of tmux.conf)
run -b ' /.tmux/plugins/tpm/tpm'

# ===== logging =====
#
# C-b C-l Start logging.
# bind-key C-l pipe-pane -o '/bin/sh -c "while read -r LINE; do echo ¥"[¥$(date
+¥"%Y_%m%d_%H%M%S_%N¥")]" ¥${LINE}¥" >> ¥${HOME}/.tmux/log/¥$(date
+¥"%m%d-%H%M%S)-#S-#l.#P.log; done "' ¥; display-message "Logging start."
# C-b C-s Stop logging
# bind-key C-s pipe-pane ¥; display-message "Logging end."
# C-b v Open new window. Logging always. (-n is set window name.)
# bind-key v new-window -n 'logging...' ¥; pipe-pane -o '/bin/sh -c "while read -r LINE; do echo
¥"[¥$(date +¥"%Y_%m%d_%H%M%S_%N¥")]" ¥${LINE}¥" >> ¥${HOME}/.tmux/log/¥$(date
+¥"%m%d-%H%M%S)-#S-#l.#P.log; done "'

```

~/.tmux.conf(以前使っていたもの)

```

# tmux list-keys

# 設定ファイルをリロードする
bind r source-file ~/.tmux.conf ¥; display "Reloaded!"

# 色が正常に表示されないことへの対応
set -g default-terminal "screen-256color"
set -g terminal-overrides 'xterm:colors=256'

# pane, window
bind -n M-p select-pane -t :-
bind -n M-@ select-pane -t :.+
bind -n M-P select-window -t :-
bind -n M-` select-window -t :+

# ウィンドウ作成
bind c new-window -c '#{pane_current_path}'
bind-key m if-shell "tmux show-options -g mouse | grep off" "set-option -g mouse on ¥; display
¥"mouse on¥" "set-option -g mouse off ¥; display ¥"mouse off¥"

# Keybind ウィンド分割
bind 3 split-window -h -c '#{pane_current_path}'
bind | split-window -h -c '#{pane_current_path}'
bind 2 split-window -v -c '#{pane_current_path}'
bind - split-window -v -c '#{pane_current_path}'

# バッファ数 ( デフォルト 2000 行 )
# set-option -g history-limit 5000

# メッセージ表示時間
set-option -g display-time 10000
set-option -g display-panes-time 10000

# マウス操作 ( コメントアウトすればターミナルのマウス操作ができるようになる )

```

```

set-option -g mouse on

#Window 名の自動設定 off
set-option -g allow-rename off

# ステータスバーをトップに配置する
# set-option -g status-position top

## ウィンドウのインデックスを 1 から始める
set -g base-index 1
## ペインのインデックスを 1 から始める
setw -g pane-base-index 1
## アクティブなペインのみ白っぽく変更 (真っ黒は 232)
set -g window-style 'bg=#212121'
set -g window-active-style 'bg=#000000'

# ステータスバー
set-option -g status-left '[#S]'
set-option -g status-right "[%Y-%m-%d %H:%M]"

# コピペ
bind v choose-buffer
set-window-option -g mode-keys vi
unbind -T copy-mode-vi MouseDragEnd1Pane
unbind-key -T copy-mode-vi Home
bind-key -T copy-mode-vi Home send -X start-of-line
unbind-key -T copy-mode-vi End
bind-key -T copy-mode-vi End send -X end-of-line
unbind-key -T copy-mode-vi C-Left
bind-key -T copy-mode-vi C-Left send -X previous-word
unbind-key -T copy-mode-vi C-Right
bind-key -T copy-mode-vi C-Right send -X next-word
unbind-key -T copy-mode-vi C-Home
bind-key -T copy-mode-vi C-Home send -X history-top
unbind-key -T copy-mode-vi C-End
bind-key -T copy-mode-vi C-End send -X history-bottom
# bind-key -T copy-mode-vi Enter send-keys -X copy-pipe "tmux copy-buffer"
bind-key -T copy-mode-vi v send-keys -X begin-selection
# for Linux
bind-key -T copy-mode-vi y send-keys -X copy-pipe-and-cancel "xclip -i -sel clip > /dev/null"
bind-key -T copy-mode-vi Enter send-keys -X copy-pipe-and-cancel "xclip -i -sel clip > /dev/null"
bind-key -T copy-mode-vi c send-keys -X copy-pipe "xclip -i -sel clip > /dev/null" ¤; display "Copy"
# for mintty
# bind-key -T copy-mode-vi y send-keys -X copy-pipe-and-cancel "cat | nkf -s | clip"
# bind-key -T copy-mode-vi Enter send-keys -X copy-pipe-and-cancel "cat | nkf -s | clip"
# bind-key -T copy-mode-vi c send-keys -X copy-pipe "cat | nkf -s | clip" ¤; display "Copy"
# for Mac
# bind-key -T copy-mode-vi y send-keys -X copy-pipe-and-cancel "pbcopy"
# bind-key -T copy-mode-vi Enter send-keys -X copy-pipe-and-cancel "pbcopy"
# bind-key -T copy-mode-vi c send-keys -X copy-pipe "pbcopy"

bind-key -T copy-mode-vi C-v send -X rectangle-toggle

# List of plugins
set -g @plugin 'tmux-plugins/tpm'
set -g @plugin 'tmux-plugins/tmux-sensible'
set -g @plugin 'tmux-plugins/tmux-copycat'

# Initialize TMUX plugin manager (keep this line at the very bottom of tmux.conf)
run -b ' /.tmux/plugins/tpm/tpm'

```

使い方

検索

選択モードで / (前方検索) または ? (後方検索)

プラグイン

プラグインマネージャ

```
git clone https://github.com/tmux-plugins/tpm /.tmux/plugins/tpm
```

~/tmux.conf

```
# List of plugins
set -g @plugin 'tmux-plugins/tpm'
set -g @plugin 'tmux-plugins/tmux-sensible'

# Initialize TMUX plugin manager (keep this line at the very bottom of tmux.conf)
run -b ' /.tmux/plugins/tpm/tpm'
```

copycat

検索で正規表現を使えるようになる。

~/.tmux.conf

```
set -g @plugin 'tmux-plugins/tmux-copycat'
```

prefix + I

自動起動

方法 1

色々あるけどとりあえず、これでいい。

~/.bashrc

```
if [[ ! -n $TMUX ]]; then
  tmux new-session && exit
fi
```

方法 2

こんな感じのシェルでもいいかも。

実行するとメニューが表示される。

キー	実行されるもの
enter	何もしない
数字	他にセッションがあるときは、セッション番号を選択してアタッチ。
n	tmux を new session で起動。
t	tmux を new session で起動。ただし、prefix を C+t にする。

```
NEW_SESSION=n
OTHER_PREFIX=t
ids="`tmux list-sessions 2> /dev/null`"

echo enter: Do Nothing
echo ${NEW_SESSION}: create new session
echo ${OTHER_PREFIX}: create new session ¥(prefix C+${OTHER_PREFIX}¥)
if [ -n "${ids}" ]; then
  echo "${ids}"
fi
read SEL

if [ -z $SEL ]; then
  : # do nothing
  exit 1
elif [ $SEL == "${OTHER_PREFIX}" ]; then
  tmux new-session¥; set-option prefix C-¥{OTHER_PREFIX}
elif [ $SEL == "${NEW_SESSION}" ]; then
  tmux new-session
else
```

```
tmux attach-session -t "${SEL}"
fi
```

ログを取る

<https://qiita.com/docokano/items/8418cc1b2561df03e8d9>

1. ログの出力先作成

1. mkdir -p ~/.tmux/log/

2. tmux.conf でログ出力を指定

```
# ===== logging =====
#
# C-b C-l Start logging.
bind-key C-l pipe-pane -o '/bin/sh -c "while read -r LINE; do echo "${date
+%"%Y-%m-%d-%H%M%S"} ${LINE}" >> ${HOME}/.tmux/log/${date
+%"%Y-%m-%d-%H%M%S"}-#S-#l.#P.log; done "' &; display-message "Logging start."
# C-b C-s Stop logging
bind-key C-s pipe-pane &; display-message "Logging end."
# C-b v Open new window. Logging always. (-n is set window name.)
bind-key v new-window -n 'logging...' &; pipe-pane -o '/bin/sh -c "while read -r LINE; do echo
${date +%"%Y-%m-%d-%H%M%S"} ${LINE}" >> ${HOME}/.tmux/log/${date
+%"%Y-%m-%d-%H%M%S"}-#S-#l.#P.log; done "'
```

1.8

```
# tmux list-keys

# 設定ファイルをリロードする
bind r source-file ~/.tmux.conf &; display "Reloaded!"

# pane, window
bind -n M-p select-pane -t :-
bind -n M-@ select-pane -t :.+
bind -n M-P select-window -t :-
bind -n M-` select-window -t :+

# Keybind ウィンド分割
bind 3 split-window -h
bind | split-window -h
bind 2 split-window -v
bind - split-window -v

# メッセージ表示時間
set-option -g display-time 10000
set-option -g display-panes-time 10000

# マウス操作
# set-window-option -g mode-mouse on
set -g mode-mouse on
set -g mouse-resize-pane on
set -g mouse-select-pane on
set -g mouse-select-window on

# Window 名の自動設定 off
# set-window-option -g automatic-rename off
set-option -g allow-rename off

# ステータスバーをトップに配置する
set-option -g status-position top

# ステータスバー
set-option -g status-left '[#S]'
set-option -g status-right "[%Y-%m-%d %H:%M]"

# コピペ
bind v choose-buffer
set-window-option -g mode-keys vi
```

その他

ssh 経由でのクリップボード共有

基本的にそのままできるはずだが、X 転送 (オプション -X) が必要。

また、Mac から Linux に SSH してクリップボードを共有する場合は、Mac 側で XQuartz のアプリを何か起動しておかないとうまくいかない時がある。

tmux の入れ子

prefix キーで入れ子の tmux を操作する

ローカルで tmux を使いつつ、リモートでも tmux を使う場合

```
ctrl + b, b
```

でリモートの tmux へ prefix を送れる

<https://qastack.jp/unix/130730/how-to-quickly-send-commands-to-nested-tmux-sessions>

また

```
# send the prefix to client inside window
bind-key -n C-a send-prefix
```

とすると ctrl + a でリモートに prefix を送れる

親の tmux の prefix を切替

Ctrl + t で、親 tmux の prefix キーの変更と prefix キー省略のショートカットの有効無効を切替える。

そうすることで、入れ子になっている親、子ともに同じショートカットで操作できる。

tmux 3.3 以降で動作可能。

.tmux.conf

```
#####
# グローバルオプション設定
#####
# ステータスバーをトップに配置する
# set -g status-position top
# set-option -g status-bg "colour51"

# バッファ数 ( デフォルト 2000 行 )
# set -g history-limit 5000

# 色が正常に表示されないことへの対応
set -g default-terminal "screen-256color"
set -g terminal-overrides 'xterm:colors=256'
# メッセージ表示時間
set -g display-time 10000
set -g display-panes-time 10000
# マウス操作 ( コメントアウトすればターミナルのマウス操作ができるようになる )
set -g mouse on
# Window 名の自動設定 off
set -g allow-rename off
# ウィンドウのインデックスを 1 から始める
set -g base-index 1
# ペインのインデックスを 1 から始める
setw -g pane-base-index 1
# アクティブなペインのみ白っぽく変更 ( 真っ黒は 232 )
set -g window-style 'bg=#212121'
set -g window-active-style 'bg=#000000'
# ステータスバー
```

```

set -g status-left '[#S]'
set -g status-right "[%Y-%m-%d %H:%M]"

#####
# クリップボード関連
#####
bind v choose-buffer
setw -g mode-keys vi
unbind -T copy-mode-vi MouseDragEnd1Pane
bind -T copy-mode-vi Home send -X start-of-line
bind -T copy-mode-vi End send -X end-of-line
bind -T copy-mode-vi C-Left send -X previous-word
bind -T copy-mode-vi C-Right send -X next-word
bind -T copy-mode-vi C-Home send -X history-top
bind -T copy-mode-vi C-End send -X history-bottom
bind -T copy-mode-vi v send-keys -X begin-selection
bind -T copy-mode-vi y send-keys -X copy-pipe-and-cancel "${COPY_CLIP}"
bind -T copy-mode-vi Enter send-keys -X copy-pipe-and-cancel "${COPY_CLIP}"
bind -T copy-mode-vi c send-keys -X copy-pipe "${COPY_CLIP}" ¥; display "Copy"
bind -T copy-mode-vi V send -X rectangle-toggle

#####
# バインド設定 (prefix)
#####
# 設定ファイルをリロードする
bind r source-file /.tmux.conf ¥; display "Reloaded!"
# ウィンドウ作成
bind c new-window -c '#{pane_current_path}'
bind m if-shell "tmux show-options -g mouse | grep off" "set-option -g mouse on ¥; display ¥"mouse
on¥"" "set-option -g mouse off ¥; display ¥"mouse off¥""
# Keybind ウィンドウ分割
bind 3 split-window -h -c '#{pane_current_path}'
bind | split-window -h -c '#{pane_current_path}'
bind 2 split-window -v -c '#{pane_current_path}'
bind - split-window -v -c '#{pane_current_path}'

#####
# バインド設定 (root)
#####
bind -T root C-t source-file /.tmux_toggle.conf
bind -n M-Pageup select-pane -t :-
bind -n M-PageDown select-pane -t :.+
bind -n C-Pageup select-window -t :-
bind -n C-PageDown select-window -t :.+
bind -n M-p select-pane -t :-
bind -n M-@ select-pane -t :.+
bind -n M-P select-window -t :-
bind -n M-` select-window -t :+

#####
# バインド設定 (disable)
#####
bind -T disable C-t source-file /.tmux_toggle.conf
# tmux list-keys -T root と合わせる
bind -T disable MouseDown1Pane select-pane -t = ¥; send-keys -M
bind -T disable MouseDown1Status select-window -t =
bind -T disable MouseDown2 Pane select-pane -t = ¥; if-shell -F
"#{||:#{mouse_in_mode},#{mouse_any_flag}}" "send -M" "paste -p"
bind -T disable MouseDown3 Pane if-shell -F -t =
"#{||:#{mouse_any_flag},#{&&:#{pane_in_mode},#{?#{m/r:(copy|view)-mode,#{pane_mode}},0,1 }}}"
"select-pane -t= ; send -M" "display-menu -t= -xM -yM -T ¥"#[align=centre]#{pane_index}
(#{pane_id})¥" ¥"#{?#{m/r:(copy|view)-mode,#{pane_mode}},Go To Top,}¥" < ¥"send -X history-top¥"
¥"#{?#{m/r:(copy|view)-mode,#{pane_mode}},Go To Bottom,}¥" > ¥"send -X history-bottom¥"
¥"#{?mouse_word,Search For #[underscore]#{=/9 /...:mouse_word,}¥" C-r ¥"if -F
¥¥¥"#{?#{m/r:(copy|view)-mode,#{pane_mode}},0,1 }¥¥¥" ¥¥¥"copy-mode -t=¥¥¥" ; send -Xt=
search-backward ¥¥¥"#{q:mouse_word}¥¥¥"¥" ¥"#{?mouse_word,Type #[underscore]#{=/9 /...:mouse_word,}¥"
C-y ¥"copy-mode -q ; send-keys -l -- ¥¥¥"#{q:mouse_word}¥¥¥"¥" ¥"#{?mouse_word,Copy
#[underscore]#{=/9 /...:mouse_word,}¥" c ¥"copy-mode -q ; set-buffer -- ¥¥¥"#{q:mouse_word}¥¥¥"¥"
¥"#{?mouse_line,Copy Line,}¥" l ¥"copy-mode -q ; set-buffer -- ¥¥¥"#{q:mouse_line}¥¥¥"¥"
¥"Horizontal Split¥" h ¥"split-window -h¥" ¥"Vertical Split¥" v ¥"split-window -v¥"
¥"#{?#{>:#{window_panes},1},,-}Swap Up¥" u ¥"swap-pane -U¥" ¥"#{?#{>:#{window_panes},1},,-}Swap
Down¥" d ¥"swap-pane -D¥" ¥"#{?pane_marked_set,,-}Swap Marked¥" s swap-pane ' Kill X kill-pane
Respawn R ¥"respawn-pane -k¥" ¥"#{?pane_marked,Unmark,Mark}¥" m ¥"select-pane -m¥"
¥"#{?#{>:#{window_panes},1},,-}#{?window_zoomed_flag,Unzoom,Zoom}¥" z ¥"resize-pane -Z¥"
bind -T disable MouseDown3Status display-menu -T "#[align=centre]#{window_index}:#{window_name}"
-t = -x W -y W "#{?#{>:#{session_windows},1},,-}Swap Left" l "swap-window -t:-1"
"#{?#{>:#{session_windows},1},,-}Swap Right" r "swap-window -t:+1" "#{?pane_marked_set,,-}Swap
Marked" s swap-window ' Kill X kill-window Respawn R "respawn-window -k"
"#{?pane_marked,Unmark,Mark}" m "select-pane -m" Rename n "command-prompt -l ¥"¥¥" ¥"rename-window
-- '%¥"¥"¥" "New After" w "new-window -a" "New At End" W new-window
bind -T disable MouseDown3StatusLeft display-menu -T "#[align=centre]#{session_name}" -t = -x M -y W
Next n "switch-client -n" Previous p "switch-client -p" ' Renumber N "move-window -r" Rename n

```

```

"command-prompt -I ¥"#S¥" ¥"rename-session -- '%%'¥" ' ' "New Session" s new-session "New Window" w
new-window
bind -T disable MouseDrag1Pane          if-shell -F "#{|:#{pane_in_mode},#{mouse_any_flag}}" "send -M"
"copy-mode -M"
bind -T disable MouseDrag1Border        resize-pane -M
bind -T disable WheelUpPane             if-shell -F "#{|:#{pane_in_mode},#{mouse_any_flag}}" "send -M"
"copy-mode -e"
bind -T disable WheelUpStatus           previous-window
bind -T disable WheelDownStatus         next-window
bind -T disable DoubleClick1 Pane       select-pane -t = ¥; if-shell -F
"#{|:#{pane_in_mode},#{mouse_any_flag}}" "send -M" "copy-mode -H ; send -X select-word ; run -d0.3 ;
send -X copy-pipe-and-cancel"
bind -T disable TripleClick1 Pane       select-pane -t = ¥; if-shell -F
"#{|:#{pane_in_mode},#{mouse_any_flag}}" "send -M" "copy-mode -H ; send -X select-line ; run -d0.3 ;
send -X copy-pipe-and-cancel"

# ===== logging =====
# C-b C-l Start logging.
# bind C-l pipe-pane -o '/bin/sh -c "while read -r LINE; do echo ¥"[¥$(date
+¥"%%Y_%%m%%d_%%H%%M%%S_%%N¥")]" ¥${LINE}¥" >> ¥${HOME}/.tmux/log/¥$(date
+¥Y%%m%%d-%%H%%M%%S)-#S-#l.#P.log; done "' ¥; display-message "Logging start."
# C-b C-s Stop logging
# bind C-s pipe-pane ¥; display-message "Logging end."
# C-b v Open new window. Logging always. (-n is set window name.)
# bind v new-window -n 'logging...' ¥; pipe-pane -o '/bin/sh -c "while read -r LINE; do echo
¥"[¥$(date +¥"%%Y_%%m%%d_%%H%%M%%S_%%N¥")]" ¥${LINE}¥" >> ¥${HOME}/.tmux/log/¥$(date
+¥Y%%m%%d-%%H%%M%%S)-#S-#l.#P.log; done "'

#####
# Plugins
#####
set -g @plugin 'tmux-plugins/tpm'
set -g @plugin 'tmux-plugins/tmux-sensible'
set -g @plugin 'tmux-plugins/tmux-copycat'

# Initialize TMUX plugin manager (keep this line at the very bottom of tmux.conf)
run -b ' /.tmux/plugins/tpm/tpm'

```

.tmux_toggle.conf

```

# display-message -p "#{session_name}"
# display-message -p "#{DISABLE}"
%if #{==:#{DISABLE},TRUE}
setenv DISABLE FALSE
set key-table root
set -u prefix
set -u status-fg
set -u status-bg
set -u key-table
%else
setenv DISABLE TRUE
set key-table disable
set prefix C-v
set status-fg white
set status-bg "colour238"
%endif

```