Terminal sharing (tmux)

To be able to see/use a terminal from multiple locations, tmux can be used to detach the terminal from the terminal client. the following configuration changes can make this happen while passing through the window title, allow for simultaneous use and show a menu with available sessions to log into while keeping the session running (ctrl-d protection).

```
sudo apt-get install tmux
```

Bashrc (or profile or zsh config):

~/.bashrc

```
# set a fancy prompt (non-color, unless we know we "want" color)
case "$TERM" in
    xterm-color|*-256color|screen*|tmux*) color_prompt=yes;;
esac

# Local desktop tmux wrapper (No-Exit Edition)
if [[ -z "$TMUX" && -z "$SSH_CONNECTION" && -n "$DISPLAY" && $- == *i*
]]; then
    # Unique session name
    SESSION_NAME="term_$(date +%s)_$$"

    # Run tmux cleanly. When you close or detach from SSH,
    # this local terminal will just sit silently instead of closing.
    tmux new-session -A -s "$SESSION_NAME" \; set-option remain-on-exit
on

    # Fallback safety: If tmux completely exits, keep the window open
    echo "Tmux session ended. Press Enter to drop to regular shell..."
    read
fi
```

tmux configuration file:

```
# Remap Ctrl+D inside tmux to DETACH instead of EXITing the shell
bind-key -n C-d detach-client

# If you actually want to kill the window/pane while sitting at your
physical desktop,
# use the native tmux shortcut: Prefix (Ctrl+B) then &

# automatically shrink/grow to match latest screen it is being looked at
setw -g window-size latest

# 1. Allow applications (like Claude) to explicitly pass-through escape
sequences
```

```
set -g allow-passthrough on

# 2. Tell tmux to actively listen to internal title changes and push them
out to xterm/gnome-terminal
set -g set-titles on

# 3. Format the actual outer title window string.
# Using #T ensures that whatever Bash (host/folder) or Claude sets gets
cleanly mirrored.
set -g set-titles-string "#T"

set -g default-terminal "xterm-256color"

# Disable status bar
set -g status off
```

Simple menu for terminal selection:

~/.local/bin/term-menu

```
#!/bin/bash

if ! tmux info &>/dev/null; then
    echo "========================================="
    echo " No Active Terminal Sessions Found"
    echo "========================================="
    echo "The tmux server is not running."
    exit 0
fi

# Fetch unique sessions using sort -u to completely eliminate double
listings
sessions=$(tmux list-sessions -F "#S" 2>/dev/null | sort -u)

if [ -z "$sessions" ]; then
    echo "No active terminal sessions found."
    exit 1
fi

echo "========================================="
echo " Active Terminal Sessions"
echo "========================================="

if_names=()
while read -r line; do
    [[ -n "$line" ]] && if_names+=("$line")
done <<< "$sessions"

for i in "${!if_names[@]}"; do
    display_title=$(tmux display-message -t "${if_names[$i]}" -p "#T")
    current_cmd=$(tmux display-message -t "${if_names[$i]}" -p
```

```
"#{pane_current_command}")

    if [ "$display_title" = "bash" ] || [ "$display_title" = "zsh" ] ||
[[ "$display_title" == *"Terminal"* ]]; then
        title_string="Running: $current_cmd"
    else
        title_string="Title: '$display_title' ($current_cmd)"
    fi

    echo "[$i] Session: ${if_names[$i]} -> $title_string"
done

echo "========================================="
read -p "Select a session number to attach (or 'q' to quit): " choice

if [[ "$choice" =~ ^[0-9]+$ ]] && [ "$choice" -lt "${#if_names[@]}" ];
then
    tmux attach-session -t "${if_names[$choice]}"
else
    echo "Exiting."
fi
```

```
chmod +x ~/.local/bin/term-menu
```

Useful commands:

```
# Reload config file
tmux source-file ~/.tmux.conf

# Kill specific session
tmux kill-session -t term_556644

# Wipe state clean/kill all sessions
tmux kill-server
```

From:
http://wuff.dyndns.org/ - **Wulf's Various Things**

Permanent link:
http://wuff.dyndns.org/doku.php?id=config:terminal-sharing&rev=1793813278

Last update: **2026/11/04 17:27**