

Утилиты-мультиплексоры screen и tmux.

Баазова Нина Эдгаровна¹

19 марта, 2024, Москва, Россия

¹Российский Университет Дружбы Народов

- Баазова Нина Эдгаровна
- студент НБИ-02-23
- Российский университет дружбы народов
- [1132239096@pfur.ru]

Изучить тему: “Утилиты-мультиплексоры screen и tmux”.

Утилиты-мультиплексоры `screen` и `tmux` являются мощными инструментами для работы в терминале в Unix-подобных системах. Они позволяют управлять несколькими сеансами терминала в одном окне, разделять окно терминала на несколько панелей, создавать окна и многое другое.

С помощью этих утилит можно эффективно работать с несколькими задачами одновременно и упростить управление окружением терминала.

- Screen – консольная утилита позволяющая в действующей SSH сессии открывать неограниченное количество независимых виртуальных терминалов (окон). Процессы, выполняемые внутри screen, не прерываются даже при обрыве сессии с сервером.
- Утилита screen является отличным инструментом как для новичков в Linux, так и для опытных системных администраторов. Внутри сессии screen можно запускать процессы, которые требуют длительное время на выполнения. В это же время вы всегда можете просто закрыть консоль и запущенный процесс будет выполняться в фоне.

- Установка screen в Fedora `yum install screen`. Запустите сессию screen одной командой `screen`. Чтобы вывести справку по использованию screen, нажмите комбинацию `ctrl + a` и после этого введите знак вопроса (?) (рис 1).

Справка по использованию

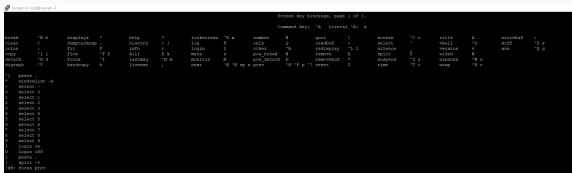
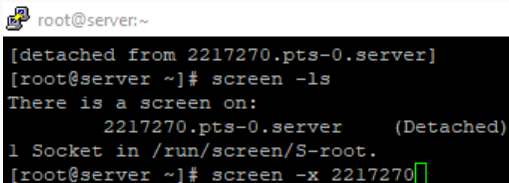


Рис. 1: Справка по использованию.

Вернуться к сессии screen можно через кнопку пробел. Чтобы полностью закрыть терминал screen, нажмите сочетание клавиш `ctrl + d`. Чтобы просто выйти из сессии screen, не закрывая ее полностью, используйте комбинацию: `Ctrl + a d`. Если у вас открыта одна сессия screen, вернуться к ней можно с помощью команды: `screen -r`. Проверить список запущенных сессий screen: `screen -list` или `screen -ls` (рис 2).

Команда screen -ls



```
root@server:~  
[detached from 2217270.pts-0.server]  
[root@server ~]# screen -ls  
There is a screen on:  
      2217270.pts-0.server      (Detached)  
1 Socket in /run/screen/S-root.  
[root@server ~]# screen -x 2217270
```

Рис. 2: Команда screen -ls.

Есть еще очень удобная функция в screen — это разделение экрана (split). Экран можно делить как по горизонтали, так и по вертикали. Например, чтобы разделить экран по горизонтали, в сессии screen нужно нажать `ctrl+a S`. Экран разделится на две части. Чтобы переключиться на вторую часть, введите `ctrl+a tab` и после переключения для активации `ctrl+a c`. Одновременно в двух частях, можно запускать нужные скрипты или утилиты (рис 3). Чтобы разделить экран по вертикали, используйте `ctrl+a`. Более того, разделителей может быть неограниченно количество (рис 4).

Запуск нескольких скриптов

```
[root@server ~]# top - 20:19:39 up 97 days, 20:04, 2 users, load average: 0.29, 0.09, 0.02
task: 114 total, 1 running, 113 sleeping, 0 stopped, 0 zombie
Cpu(s): 11.5 us, 12.3 sy, 9.5 ni, 59.7 id, 2.1 wa, 0.7 hi, 1.8 si, 2.5 st
Mem: 1826.8 total, 293.6 free, 216.1 used, 1317.0 buff/cache
Mem Swap: 0.0 total, 0.0 free, 0.0 used, 1336.8 avail Mem

  PID USER      SI  BL  VIRT  RSS  SHR  S#    %CPU  %MEM     time+  command
 113 root      20    0 179118 10782 12128 S  0.7  0.6  10:17:13 systemd
 713 root      20    0 382044 17648 11380 S  4.7  0.9  4462:13 NetworkManager
 468 root      20    0 74828 1876 1040 S  0.3  0.0  25:10:40 dnsmasq
 10 root      20    0 0 0 0 S  0.3  0.0 150:52.11 rcu_sched
 17 root      20    0 0 0 0 S  0.3  0.0 30:17.31 ksoftirqd/1
 704 root      20    0 225069 61540 36548 S  0.3  2.2 35:01.52 ssd nas
 760 root      20    0 84340 7716 4652 S  0.3  0.0 0:58.33 systemd-logind
2219828 root  20    0 41281 3200 2200 S  0.3  0.2 0:00.04 screen
2219849 root  20    0 23480 4064 3556 S  0.3  0.2 0:00.19 bash
 2 root      20    0 0 0 0 S  0.0  0.0 0:00.55 kthreadd
 3 root      0 ~20 0 0 0 S  0.1  0.0 0:00.00 rcu_gp
 4 root      0 ~20 0 0 0 S  0.1  0.0 0:00.00 rcu_paw_gp
 6 root      0 ~20 0 0 0 S  0.0  0.0 0:00.00 kworker/1:0G-@kiockd
 8 root      0 ~20 0 0 0 S  0.1  0.0 0:00.00 mm_percpu_wq
 9 root      20    0 0 0 0 S  0.0  0.0 0:00.44 ksoftirqd/0
11 root      rx  0 0 0 S  0.0  0.0 0:00.72 migration/0
12 root      rx  0 0 0 S  0.0  0.0 0:04.56 watchdog/0
13 root      20    0 0 0 0 S  0.0  0.0 0:00.00 cpufreqd
14 root      20    0 0 0 0 S  0.0  0.0 0:00.00 cpufreqd/1
15 root      rx  0 0 0 S  0.0  0.0 0:04.55 watchdog/1
16 root      rx  0 0 0 S  0.0  0.0 0:04:29 migration/1
18 root      0 ~20 0 0 0 S  0.1  0.0 0:00.00 kworker/1:0G-@kiockd
21 root      20    0 0 0 0 S  0.0  0.0 0:00.00 kdevfsd

0 packages:
Package Architecture Version
Installing:
stop x86_64 2.4.0-4.el10
Transaction Summary
Install 1 Package
Total download size: 187 K
Installed size: 432 K
Is this ok [Y/n]: y
Downloading Packages:
stop-2.4.0-4.el10.x86_64.rpm
Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing :
Installing : stop-2.4.0-4.el10.x86_64
Running scriptlet: stop-2.4.0-4.el10.x86_64
Verifying : stop-2.4.0-4.el10.x86_64
Installed:
  stop-2.4.0-4.el10.x86_64
Complete!
[root@server ~]#
```

Рис. 3: Запуск нескольких скриптов.

Пример разделения окон

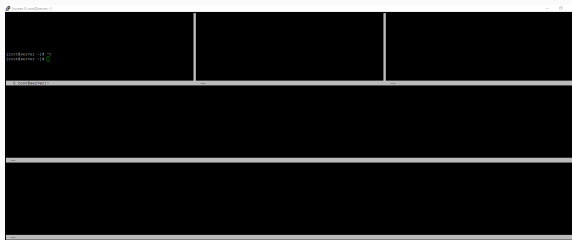


Рис. 4: Пример разделения окон.

Основные команды для работы с утилитой Screen:

- `screen` - запуск нового сеанса
- `Ctrl+a d` - отсоединиться от текущего сеанса
- `screen -r` - подключиться к отсоединенному сеансу

Tmux - это более современный аналог утилиты Screen, который обладает богатым набором функций и возможностей для работы с терминалом. Tmux также позволяет создавать несколько сеансов, окон и панелей, при этом предоставляя более гибкие настройки и возможности для пользователей (рис 5).

Установка Tmux в Fedora `dnf install tmux`. Команда запуска `tmux`.

Примеры сеансов Тmux.

```
bad averages: 0.45, 0.39, 0.37
no processes: 80 idle, 1 on processor
CPU state: 0.0% user, 0.0% nice, 0.0% system, 0.0% interrupt, 100% idle
CPU1 state: 0.0% user, 0.0% nice, 0.0% system, 0.0% interrupt, 100% idle
Memory: Real: 6897204K act/tot, free: 3668K Swap: 0/20520 used/tot

PID USERNAME PRI NICE SIZE RES STATE TIME CPU COMMAND
26300 nicholas 2 0 1776K 4780K sleep/1 poll 0.06 0.00% npd
16266 nicholas 2 0 1528K 4556K sleep/1 poll 1.34 0.00% npd
22280 nicholas 2 0 4172K 2944K sleep/0 poll 0.00 0.00% npd
2790 nicholas 2 0 3368K 1852K sleep/1 poll 0.00 0.00% scmpo
12060 root 2 0 456K 796K sleep/0 kread 0.00 0.00% apmd
7401 user 2 0 1548K 2548K sleep/1 select 0.00 0.00% htpd
38926 root 2 0 1124K 2184K sleep/1 select 0.00 0.00% sendmail
6824 root 2 0 1184K 1168K sleep/1 poll 0.01 0.00% logfdm
15382 nicholas 2 0 3384K 2268K sleep/0 select 0.02 0.00% send
1680 root 2 0 148K 148K idle nfad 0.02 0.00% nfad
36598 root 2 0 148K 148K idle nfad 0.01 0.00% nfad
76 nicholas 2 0 1304K 2124K sleep/0 poll 0.00 0.00% tmux
38891 root 2 0 612K 952K idle select 0.00 0.00% cron
18540 nicholas 3 0 600K 628K idle tynin 0.00 0.00% kbh
12971 syslogd 2 0 624K 840K sleep/0 poll 0.00 0.00% syslogd
19861 nicholas 2 0 972K 2784K sleep/1 poll 0.00 0.00% ncmpo
27353 nicholas 2 0 1588K 131K sleep/0 select 0.00 0.00% emacs

client_req_in_detach(struct hdr *hdr, struct client_ctx *ctx)
client_req_in_detach(struct lmq *lmq, struct client_ctx *ctx)
{
    if (hdr->size != 0)
        if (lmq->hdr->len != MSG_HEADER_SIZE)
            fatalx("bad MSG_DETACH size");
    client_write_server(ctx, MSG_EXITING, NULL, 0);
}
-96,9 +187,9 00

int
client_req_in_shutdown
(struct hdr *hdr, struct client_ctx *ctx)
{
    struct lmq *lmq; struct client_ctx *ctx;
    if (hdr->size != 0)
        if (lmq->hdr->len != MSG_HEADER_SIZE)
            fatalx("bad MSG_SHUTDOWN size");
    client_write_server(ctx, MSG_EXITING, NULL, 0);
}
-180,9 +112,9 00

nicholas@pinus 0 1 % is tmux+
tmux-bd-auth.diff tmux-neusetopt.diff
tmux-cfgcur.diff tmux-neusetopt1.diff
tmux-lmq-12diff.diff tmux-print.diff
tmux-insgl.diff tmux-session-new-old.diff
tmux-insgl.diff tmux-session-new.diff
tmux-mode+search.diff tmux-visual.diff
nicholas@pinus 0 1 %

client_req_in_detach(struct hdr *hdr, struct client_ctx *ctx)
client_req_in_detach(struct lmq *lmq, struct client_ctx *ctx)
{
    if (hdr->size != 0)
        if (lmq->hdr->len != MSG_HEADER_SIZE)
            fatalx("bad MSG_DETACH size");
    client_write_server(ctx, MSG_EXITING, NULL, 0);
}
-96,9 +187,9 00

int
client_req_in_shutdown
(struct hdr *hdr, struct client_ctx *ctx)
{
    struct lmq *lmq; struct client_ctx *ctx;
    if (hdr->size != 0)
        if (lmq->hdr->len != MSG_HEADER_SIZE)
            fatalx("bad MSG_SHUTDOWN size");
    client_write_server(ctx, MSG_EXITING, NULL, 0);
}
-180,9 +112,9 00

nicholas@pinus 0 1 % diff 174 (124,0) - 174 (0,0) diff/*****
nicholas@pinus 0 1 %
```

Рис. 5: Примеры сеансов Тmux.

Новое окно (нажать CTRL+b, затем нажать c) .

Переключение

- // следующее окно
- // предыдущее окно
- <C-b 0> // переключиться на номер окна

Деление окна горизонтально <C-b "> или `tmux split-window -h` (рис 6).

Деление окна вертикально <C-b %> или `tmux split-window -v` (рис 7).

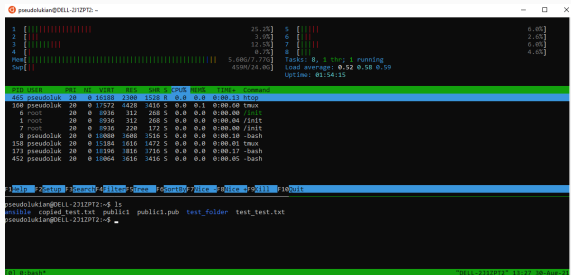
Заккрытие окон // нужно подтвердить у или exit.

Отключение от сессии или `tmux detach`.

Список сессий `tmux ls`.

Выбрать сессию (рис 8).

Пример деления окон горизонтально



```
pseudokian@DELL-2312PT2:~$ top
top - 09:54:15
 1  [|||||] 25.2% 5 [|||||] 6.0%
 2  [|||||] 3.0% 6 [|||||] 2.6%
 3  [|||||] 12.5% 7 [|||||] 6.0%
 4  [|||||] 0.7% 8 [|||||] 0.1%
Mem: 5.60G/7.77G Tasks: 0, 1 thr: 1 running
Swap: 459M/24.0G load average: 0.52 0.58 0.58
                                         uptime: 01:54:15

 PID USER      PR  NI  VIRT  RES  SHR  S#%C   TIME Command
 465 pseudok 20   0 16188 2300 1528 R 0.0  0.0 0:00.13 httpd
160 pseudok 20   0 17572 4428 3416 S 0.0  0.1 0:00.00 tmux
 6 root      20   0 8936 312 268 S 0.0  0.0 0:00.00 init
 1 root      20   0 8936 312 268 S 0.0  0.0 0:00.04 /init
 7 root      20   0 8936 220 172 S 0.0  0.0 0:00.00 /init
 8 pseudok 20   0 18080 568 3512 S 0.0  0.0 0:00.18 -bash
158 pseudok 20   0 15184 1616 1472 S 0.0  0.0 0:00.01 tmux
173 pseudok 20   0 18196 3816 3716 S 0.0  0.0 0:00.17 -bash
452 pseudok 20   0 18064 3616 3416 S 0.0  0.0 0:00.05 -bash

pseudokian@DELL-2312PT2:~$
pseudokian@DELL-2312PT2:~$ ls
visible  copied_test.txt  public1  public1.pub  test_folder  test_test.txt
pseudokian@DELL-2312PT2:~$
```

Рис. 6: Пример деления окон горизонтально.

Пример деления окон вертикально

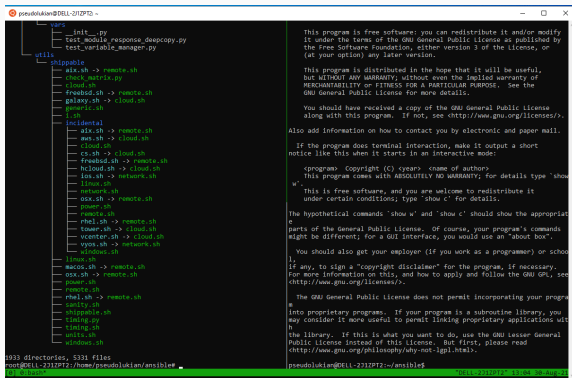
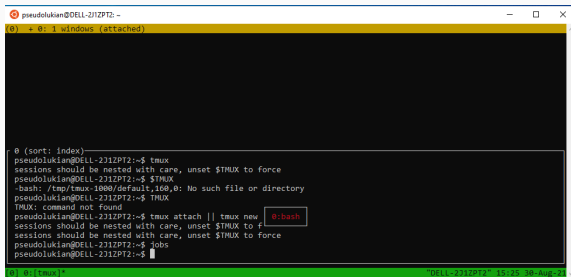


Рис. 7: Пример деления окон вертикально.

Пример выбора сессии



```
pseudolukian@DELL-211ZPT2: -  
0) + 0: 1 windows (attached)  
  
0 (sort: index)  
pseudolukian@DELL-211ZPT2:~$ tmux  
sessions should be nested with care, unset $TMUX to force  
pseudolukian@DELL-211ZPT2:~$ $TMUX  
-bash: /tmp/tmux-1000/default,160,0: No such file or directory  
pseudolukian@DELL-211ZPT2:~$ $TMUX  
tmux: command not found  
pseudolukian@DELL-211ZPT2:~$ tmux attach || tmux new  
sessions should be nested with care, unset $TMUX to f  
sessions should be nested with care, unset $TMUX to force  
pseudolukian@DELL-211ZPT2:~$ jobs  
pseudolukian@DELL-211ZPT2:~$
```

Рис. 8: Пример выбора сессии.

Основные команды для работы с утилитой Tmux:

- `tmux` - запустить новый сеанс
- `Ctrl+b d` - отсоединиться от текущего сеанса
- `tmux attach` - подключиться к отсоединенному сеансу

Обе утилиты предоставляют аналогичные функциональные возможности, но Tmux имеет более продвинутый и гибкий интерфейс, а также поддерживает более сложные конфигурации окон и панелей. Screen является более простым в использовании и имеет более широкое распространение из-за своей долгой истории.

Утилиты-мультиплексоры Screen и Tmux являются незаменимыми инструментами для работы в терминале, позволяя повысить производительность и эффективность работы с командной строкой. Выбор между Screen и Tmux будет зависеть от индивидуальных предпочтений и требований пользователя, однако обе утилиты заслуживают внимания благодаря своей функциональности и удобству использования.

Мы изучили тему: “Утилиты-мультиплексоры screen и tmux”.

Ссылки на источники:

1. Использование утилиты screen в Linux
2. Как пользоваться утилитой screen в Linux?
3. Что такое tmux
4. Шпаргалка по работе с Tmux
5. Шпаргалка по tmux
6. Утилита-мультиплексор tmux
7. Шпаргалка по Tmux: установка, настройка, работа.