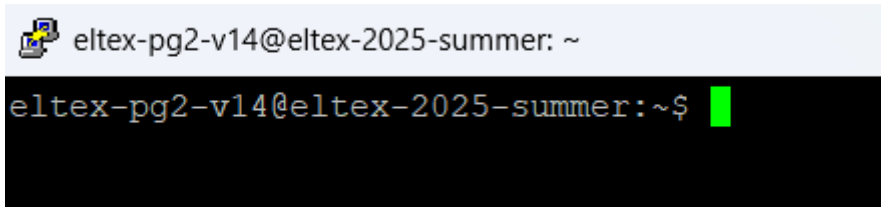


1. Запустите новую сессию с именем "'whoami'_part2". Все команды сохраняйте в файле \$USER_part.log, проследите за тем, чтобы он не содержал есc-последовательности форматирования и расцветки.

Пытался сделать автоматическое логирование, не уверен, что получилось

```
screen -S "'whoami'_part2" -L -Logfile $USER_part.log
```

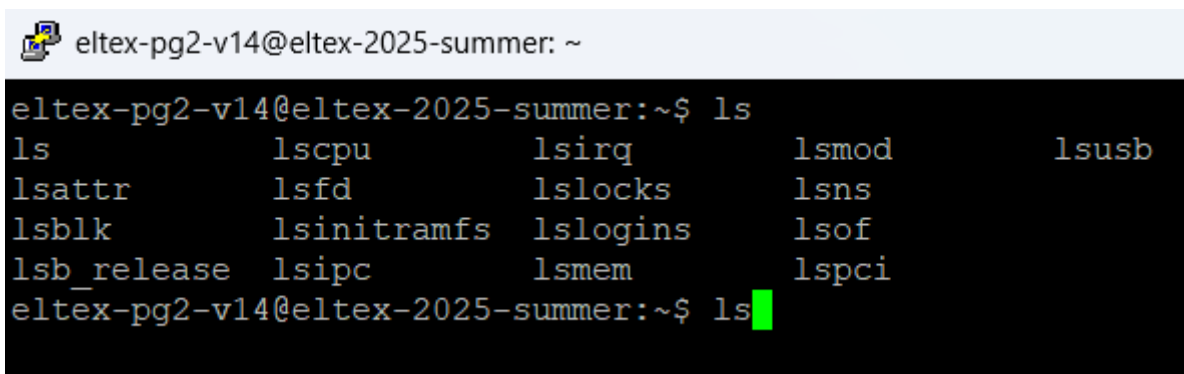


```
eltex-pg2-v14@eltex-2025-summer: ~$
```

Позже выяснилось, что он просто .log, потому что нужно было написать \${USER}

2. С помощью механизма дополнения имен команд выведите все команды, которые начинаются на «ls».

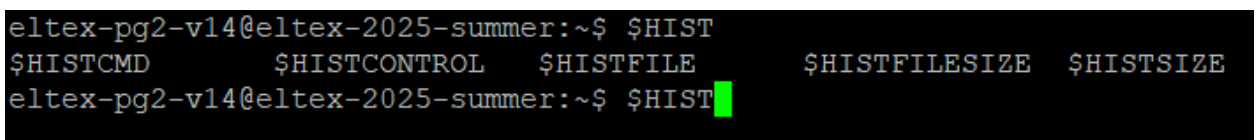
ls + два раза нажать таб



```
eltex-pg2-v14@eltex-2025-summer: ~$ ls
ls          lscpu      lsirq      lsmod      lsusb
lsattr      lsfd       lslocks    lsns
lsblk       lsinitramfs lslogins    lsof
lsb_release lsipc      lsmem      lspci
eltex-pg2-v14@eltex-2025-summer: ~$ ls
```

3. С помощью механизма дополнения имен переменных выведите все переменные, которые начинаются с «HIST».

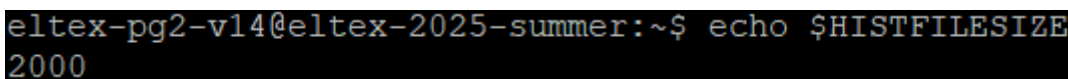
\$HIST + 2 TAB



```
eltex-pg2-v14@eltex-2025-summer: ~$ $HIST
$HISTCMD      $HISTCONTROL  $HISTFILE      $HISTFILESIZE  $HISTSIZE
eltex-pg2-v14@eltex-2025-summer: ~$ $HIST
```

4. Узнайте, сколько команд может храниться в файле истории.

echo \$HISTFILESIZE



```
eltex-pg2-v14@eltex-2025-summer: ~$ echo $HISTFILESIZE
2000
```

5. Выведите имена файлов и каталогов из домашнего каталога, которые начинаются с «.».

ls -d .* символы подстановки ? – один непустой символ, * - хоть сколько символов

```
eltex-pg2-v14@eltex-2025-summer:~$ ls -d .*?  
.bash_history .bash_logout .bashrc .config .log .profile .viminfo  
eltex-pg2-v14@eltex-2025-summer:~$
```

```
eltex-pg2-v14@eltex-2025-summer:~$ ls .*?  
.bash_history .bash_logout .bashrc .log .profile .viminfo  
  
.config:  
procps  
eltex-pg2-v14@eltex-2025-summer:~$
```

6. Настройте вывод даты выполнения команд, хранящихся в истории.

```
export HISTTIMEFORMAT='%d.%m.%Y %H:%M:%S '
```

history

```
52 10.10.2025 12:35:22 ls  
53 10.10.2025 12:35:22 echo "export LANG=re_RU.UTF8">> ~/.bashrc  
54 10.10.2025 12:35:22 exit  
55 10.10.2025 12:41:54 echo $HISTFILESIZE  
56 10.10.2025 12:46:42 echo ~/.*  
57 10.10.2025 12:51:47 ls  
58 10.10.2025 12:52:09 ls -a  
59 10.10.2025 12:53:57 ls -d .*?  
60 10.10.2025 12:54:05 ls -ld .*?  
61 10.10.2025 12:54:08 ls -d .*?  
62 10.10.2025 13:09:23 ls .*?  
63 10.10.2025 13:11:10 date  
64 10.10.2025 13:12:36 -history  
65 10.10.2025 13:12:59 echo /.bash_history  
66 10.10.2025 13:13:43 echo $HISTFILE  
67 10.10.2025 13:14:21 vi /.bash_history  
68 10.10.2025 13:14:36 ls  
69 10.10.2025 13:14:58 vi ~/.bash_history  
70 10.10.2025 13:16:12 ls  
71 10.10.2025 13:16:41 history  
72 10.10.2025 13:18:19 export HISTTIMEFORMAT='%d.%m.%Y %H:%M:%S '  
73 10.10.2025 13:18:23 history  
eltex-pg2-v14@eltex-2025-summer:~$
```

0°C
Облачно



ENG



13:19
10.10.2025

7. Настройте автоматическое сохранение набираемых команд в файле истории:

- введите любую команду, например, команду date;

```
eltex-pg2-v14@eltex-2025-summer:~$ date  
Fri Oct 10 13:19:54 +07 2025
```

- проверьте, есть ли эта команда в кэше и файле истории команд;

```
74 10.10.2025 13:19:54 date  
75 10.10.2025 13:20:16 history  
eltex-pg2-v14@eltex-2025-summer:~$
```

- определить переменную PROMPT_COMMAND так, чтобы кэш истории сохранялся в файле истории;

```
eltex-pg2-v14@eltex-2025-summer:~$ export PROMPT_COMMAND='history -a'
```

- ввести любую команду и проверить, появилась ли эта команда в кэше и файле истории.

```
78 10.10.2025 13:25:12 date
79 10.10.2025 13:25:15 history
eltex-pg2-v14@eltex-2025-summer:~$
```

```
history
#1760077194
date
#1760077216
history
#1760077285
man history
#1760077493
export PROMPT_COMMAND='history -a'
#1760077512
date
#1760077515
history
~
~
```

10. Создайте переменную DATE, в которую запишите текущую дату. Проверьте содержимое переменной.

DATE = \$(date) СКОБКИ!!!

```
~
eltex-pg2-v14@eltex-2025-summer:~$ DATE = $date
bash: DATE: command not found
eltex-pg2-v14@eltex-2025-summer:~$ DATE=$date
eltex-pg2-v14@eltex-2025-summer:~$ echo $DATE

eltex-pg2-v14@eltex-2025-summer:~$ DATE=$(date)
eltex-pg2-v14@eltex-2025-summer:~$ echo $DATE
Fri Oct 10 13:29:04 +07 2025
eltex-pg2-v14@eltex-2025-summer:~$
```

DATE=\$(date +%d.%m.%Y) Даже так

```
eltex-pg2-v14@eltex-2025-summer:~$ DATE=$(date +%d.%m.%Y)
eltex-pg2-v14@eltex-2025-summer:~$ echo $DATE
10.10.2025
eltex-pg2-v14@eltex-2025-summer:~$
```

11. Создайте переменную TIME, в которую запишите текущее время. Проверьте содержимое переменной.

```
TIME=$(date +%H:%M:%S)
```

```
eltex-pg2-v14@eltex-2025-summer:~$ TIME=$(date +%H:%M:%S)
eltex-pg2-v14@eltex-2025-summer:~$ echo $TIME
13:39:15
```

12. Создайте переменную DATE_TIME в которую поместите значения из переменных DATE и TIME, разделенных пробелом. Проверьте содержимое переменной.

```
DATE_TIME="$DATE $TIME"
```

```
eltex-pg2-v14@eltex-2025-summer:~$ DATE_TIME="$DATE $TIME"
eltex-pg2-v14@eltex-2025-summer:~$ echo DATE_TIME
DATE_TIME
eltex-pg2-v14@eltex-2025-summer:~$ echo $DATE_TIME
10.10.2025 13:39:15
eltex-pg2-v14@eltex-2025-summer:~$
```

13. Выведите имена файлов, содержащие хотя бы одну цифру, из каталогов /bin и /sbin.

```
eltex-pg2-v14@eltex-2025-summer:~$ find /usr/bin -name "[0123456789]*"
```

```
/usr/bin/pdb3.11
/usr/bin/perl5.36-x86_64-linux-gnu
/usr/bin/c99
/usr/bin/x86_64-linux-gnu-strings
/usr/bin/linux64
/usr/bin/x86_64-linux-gnu-gp-display-html
/usr/bin/pygettext3.11
/usr/bin/systemd-id128
/usr/bin/x86_64-linux-gnu-dwp
/usr/bin/gcov-12
/usr/bin/i386
/usr/bin/bunzip2
/usr/bin/cpp-12
/usr/bin/x86_64-linux-gnu-gp-display-text
/usr/bin/x86_64-linux-gnu-addr2line
/usr/bin/py3compile
/usr/bin/grub-syslinux2cfg
/usr/bin/traceroute6
/usr/bin/diff3
/usr/bin/gcc-nm-12
/usr/bin/gcc-ar-12
/usr/bin/sha256sum
eltex-pg2-v14@eltex-2025-summer:~$
```

```
eltex-pg2-v14@eltex-2025-summer:~$ find /usr/sbin -name "[0123456789]*"
```

```
/usr/sbin/dumpe2fs
/usr/sbin/e2fsck
/usr/sbin/mkfs.ext3
/usr/sbin/e2freefrag
/usr/sbin/e2undo
/usr/sbin/e2scrub_all
/usr/sbin/tune2fs
/usr/sbin/e2image
/usr/sbin/e2label
/usr/sbin/e4defrag
/usr/sbin/e2mmpstatus
/usr/sbin/mke2fs
/usr/sbin/mkfs.ext4
/usr/sbin/mkfs.ext2
/usr/sbin/e4crypt
/usr/sbin/killall5
/usr/sbin/fsck.ext3
/usr/sbin/fsck.ext2
/usr/sbin/e2scrub
/usr/sbin/resize2fs
/usr/sbin/fsck.ext4
/usr/sbin/update-grub2
```

14. Измените приглашение так, чтобы выводились имя хоста, имя пользователя и время: имя_пользователя@имя_хоста-НН:ММ> (Используйте переменные bash и команду date)

```
eltex-pg2-v14@eltex-2025-summer:~$ export PS1='\u@\h-$(date +%H:%M) '
eltex-pg2-v14@eltex-2025-summer-13:55export PS1='\u@\h-$(date +%H:%M) : '
eltex-pg2-v14@eltex-2025-summer-13:56:export PS1='\u@\h-$(date +%H:%M)>'
eltex-pg2-v14@eltex-2025-summer-13:56>
```

15. Сделайте так, чтобы в запускаемом интерпретаторе bash выводилось приглашение, установленное в родительском интерпретаторе bash.

```
eltex-pg2-v14@eltex-2025-summer-13:56>bash
eltex-pg2-v14@eltex-2025-summer:~$ echo $PS1
${debian_chroot:+($debian_chroot)}\u@\h:\w$
eltex-pg2-v14@eltex-2025-summer:~$
```

"export PS1='\u@\h-\$(date +%H:%M)>'>> ~/.bashrc

```
eltex-pg2-v14@eltex-2025-summer:~$ echo "export PS1='\u@\h-$(date +%H:%M) '" >>
~/.bashrc
eltex-pg2-v14@eltex-2025-summer:~$ bash
eltex-pg2-v14@eltex-2025-summer-14:02echo "export PS1='\u@\h-$(date +%H:%M)>' "
>> ~/.bashrc
eltex-pg2-v14@eltex-2025-summer-14:02bash
eltex-pg2-v14@eltex-2025-summer-14:03>
```

16. Одной командной строкой создайте в домашнем каталоге подкаталоги для каждого месяца текущего года вида YYYY-MM(год реализуйте с помощью команды date и командной подстановки).

mkdir ~/\$(date +%Y)-{01,02,03,04,05,06,07,08,09,10,11,12}

```
eltex-pg2-v14@eltex-2025-summer-14:03>ls
report_part.txt
eltex-pg2-v14@eltex-2025-summer-14:03>mkdir ~/$(date +%Y)-{01,02,03,04,05,06,0
7,08,09,10,11,12}
eltex-pg2-v14@eltex-2025-summer-14:03>ls
2025-01 2025-03 2025-05 2025-07 2025-09 2025-11 report_part.txt
2025-02 2025-04 2025-06 2025-08 2025-10 2025-12
eltex-pg2-v14@eltex-2025-summer-14:03>
```