

ДИСЦИПЛИНА	Технологии разработки серверных приложений
ИНСТИТУТ	Институт перспективных технологий и индустриального программирования
КАФЕДРА	Индустриального программирования
ВИД УЧЕБНОГО МАТЕРИАЛА	Практическая работа
ПРЕПОДАВАТЕЛЬ	Макиевский Станислав Евгеньевич
СЕМЕСТР	4 семестр, 2024-2025



Цель работы: научиться создавать и запускать сценарии с помощью командного интерпретатора `bash`.

Задание:

0. Регулярные выражения

Схема записи регулярных выражений:

<code>^</code>	начало строки	<code>*</code>	предыдущий символ повторяется 0 и более раз
<code>\$</code>	конец строки	<code>+</code>	предыдущий символ повторяется 1 и более раз
<code>.</code>	любой символ	<code>?</code>	предыдущий символ повторяется 0 или 1 раз
<code>[символы]</code>	один символ из диапазона	<code>{n,m}</code>	предыдущий символ повторяется от n до m раз
<code>[^символы]</code>	один символ из не диапазона	<code>{n,}</code>	предыдущий символ повторяется n и более раз
<code>символы символы</code>	или	<code>{n}</code>	предыдущий символ повторяется n раз
<code>(символы)</code>	группировка символов	<code>\</code>	отмена специального значения последующего метасимвола
<code>\<слово></code>	слово		

1. Создание и запуск простых скриптов, использование конструкции `if then`

1.1. Создайте пустой файл с произвольным именем. В первой строке укажите, какая именно оболочка будет использована для обработки сценария.

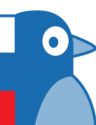
```
touch myscript
#!/bin/bash
```

Примечание: в других строках этого файла символ решетки используется для обозначения комментариев, которые оболочка не обрабатывает. Однако, первая строка – это исключение, здесь далее следует восклицательный знак и указывается путь к `bash`.

```
student@prac-work-question:~#
```

Предоставьте ответ в виде скриншота(-ов), где каждый шаг (действие) сопровождается письменным описанием.

Если необходимо предоставить скрипт, то ответ может содержать ссылку на скрипт решения (только `git`):



```
lylin@astra:~$ touch myscript
lylin@astra:~$ #!/bin/bash
lylin@astra:~$ █
```

1.2. Добавьте в скрипт команды, которые будут выводить текущую директорию и пользователя.

```
echo "Текущая директория:"
pwd
echo "Пользователь:"
whoami
```

student@prac-work-question:~# Предоставьте ответ в виде скриншота(-ов), где каждый шаг (действие) сопровождается письменным описанием.

Если необходимо предоставить скрипт, то ответ может содержать ссылку на скрипт решения (только git):

```
lylin@astra:~$ echo "Текущая директория:"
Текущая директория:
lylin@astra:~$ pwd
/home/lylin
lylin@astra:~$ echo "Пользователь:"
Пользователь:
lylin@astra:~$ whoami
lylin
lylin@astra:~$ █
```

1.3. Сделайте файл исполняемым и запустите сценарий.

```
chmod +x myscript
./myscript
```



student@prac-work-question:~# Предоставьте ответ в виде скриншота(-ов), где каждый шаг (действие) сопровождается письменным описанием.

Если необходимо предоставить скрипт, то ответ может содержать ссылку на скрипт решения (только git):

```
lylin@astra:~$ chmod +x myscript
lylin@astra:~$ ./myscript
lylin@astra:~$
```

1.4. Сделайте так, чтобы текущая директория и пользователь выводились с помощью переменных окружения.

```
echo "Текущая директория: $PWD"
echo "Пользователь: $USER"
```

student@prac-work-question:~# Предоставьте ответ в виде скриншота(-ов), где каждый шаг (действие) сопровождается письменным описанием.

Если необходимо предоставить скрипт, то ответ может содержать ссылку на скрипт решения (только git):

```
lylin@astra:~$ echo "Текущая директория: $PWD"
Текущая директория: /home/lylin
lylin@astra:~$ echo "Пользователь: $USER"
Пользователь: lylin
lylin@astra:~$
```



1.5. Добейтесь такого же результата с помощью объявления собственных переменных.

Примечание: одна из самых полезных возможностей bash-скриптов – это возможность извлекать информацию из вывода команд и назначать ее переменным, что позволяет использовать эту информацию в файле сценария. Сделать это можно двумя способами:

- 1) С помощью значка обратного апострофа «`»
- 2) С помощью конструкции `$()`

```
mydir=`pwd`  
myuser=$(whoami)  
echo "Текущая директория: $mydir"  
echo "Пользователь: $myuser"
```

student@prac-work-question:~# Предоставьте ответ в виде скриншота(-ов), где каждый шаг (действие) сопровождается письменным описанием.

Если необходимо предоставить скрипт, то ответ может содержать ссылку на скрипт решения (только git):

```
lylin@astra:~$ mydir=`pwd`  
lylin@astra:~$ myuser=$(whoami)  
lylin@astra:~$ echo "Текущая директория: $mydir"  
Текущая директория: /home/lylin  
lylin@astra:~$ echo "Пользователь: $myuser"  
Пользователь: lylin  
lylin@astra:~$
```

1.6. Дополните сценарий: необходимо найти пользователя `user1` в `/etc/passwd`. Если его удалось найти, то сообщить о том, что он существует. Если нет – вывести сообщение о том, что такого пользователя не существует.

```
user=user1  
if grep $user /etc/passwd  
then  
echo "Пользователь $user найден"  
else  
echo "Пользователь $user не найден"  
fi
```



student@prac-work-question:~# Предоставьте ответ в виде скриншота(-ов), где каждый шаг (действие) сопровождается письменным описанием.

Если необходимо предоставить скрипт, то ответ может содержать ссылку на скрипт решения (только git):

```
lylin@astra:~$ user=user1
lylin@astra:~$ if grep $user /etc/passwd
> then
> echo "Пользователь $user найден"
> else
> echo "Пользователь $user не найден"
> fi
user1:x:1500:1500::/home/user1:/bin/bash
"Пользователь user1 найден"
lylin@astra:~$
```

1.7. Проверьте существование директории Debian, перейдите в нее и выведите ее содержимое на экран.

```
dir=~ /Debian
if [ -d $dir ]
then
echo "Директория $dir существует"
cd $dir
ls
else
echo "Директория $dir не существует"
fi
```

Примечание: следующие конструкции проверяют различные условия, касающиеся файлов.

- d file – существует ли файл, и является ли он директорией;
- e file – существует ли файл;
- f file – существует ли файл, и является ли он файлом;
- r file – существует ли файл, и доступен ли он для чтения;
- s file – существует ли файл, и не является ли он пустым;
- w file – существует ли файл, и доступен ли он для записи;
- x file – существует ли файл, и является ли он исполняемым;
- file1 -nt file2 – проверяет, новее ли file1, чем file2;
- file1 -ot file2 – проверяет, старше ли file1, чем file2;
- O file – проверяет, существует ли файл, и является ли его владельцем

текущий пользователь;



-G file – проверяет, существует ли файл, и соответствует ли его идентификатор группы идентификатору группы текущего пользователя.

student@prac-work-question:~# Предоставьте ответ в виде скриншота(-ов), где каждый шаг (действие) сопровождается письменным описанием.

Если необходимо предоставить скрипт, то ответ может содержать ссылку на скрипт решения (только git):

```
lylin@astra:~$ dir=~ /Debian
lylin@astra:~$ if [ -d $dir ]
> then
> echo "Директория $dir существует"
> cd $dir
> ls
> else
> echo "Директория $dir не существует"
> fi
"Директория /home/lylin/Debian не существует"
lylin@astra:~$
```

1.8. Усложните сценарий: проверьте существование директории DBMS и файла file1 в директории DBMS/MongoDB. Если они существуют, то вывести содержимое директории DBMS и файла file1. Если нет, вывести сообщение.

```
dir1=~ /DBMS
file=~ /DBSM/MongoDB/file1
if [ -d $dir1 ] && [ -f $file ]
then
echo "Директория $dir1 и файл $file существуют"
cd $dir1 && ls && cat $file
elif [ -d $dir1 ] || [ -f $file ]
then "Директория $dir1 не существует или файл $file не существует"
else
echo "Директория $dir1 не существует и файл $file не существует"
fi
```

student@prac-work-question:~# Предоставьте ответ в виде скриншота(-ов), где каждый шаг (действие) сопровождается письменным описанием.

Если необходимо предоставить скрипт, то ответ может содержать ссылку на скрипт решения (только git):



```

lylin@astra:~$ dir1=~ /DBMS
lylin@astra:~$ file=~ /DBSM/MongoDB/file1
lylin@astra:~$ if [ -d $dir1 ] && [ -f $file ]
> then
> echo "Директория $dir1 и файл $file существуют"
> cd $dir1 && ls && cat $file
> elif [ -d $dir1 ] || [ -f $file ]
> then "Директория $dir1 не существует или файл $file не существует"
> else
> echo "Директория $dir1 не существует и файл $file не существует"
> fi
bash: "Директория: команда не найдена
lylin@astra:~$ █

```

2. Циклы for

IFS (Internal Field Separator) позволяет указывать разделители полей для сценария. По умолчанию оболочка `bash` считает разделителями полей следующие символы:

- пробел;
- знак табуляции;
- знак перевода строки.

Если `bash` встречает в данных любой из этих символов, он считает, что перед ним находится следующее самостоятельное значение списка.

Для того, чтобы решить проблему, можно временно изменить переменную среды `IFS`. Если необходимо, чтобы в качестве разделителя полей использовался только перевод строки, используется конструкция: `IFS=$'\n'`.

2.1. Рассмотрите разницу между выводом файла `/etc/passwd`, где разделителем является знак перевода строки и знак двоеточие.

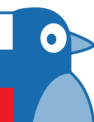
```

file1=/etc/passwd
IFS=$'\n'
for var in $(cat $file1)
do
echo $var
done

```

Далее измените `IFS=$'\n'` на `IFS=:`

В этом примере задействована команда `cat`, которая читает содержимое файла. Полученный список значений передается в цикл и выводится на экран.



student@prac-work-question:~#

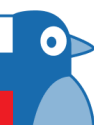
Предоставьте ответ в виде скриншота(-ов), где каждый шаг (действие) сопровождается письменным описанием.

Если необходимо предоставить скрипт, то ответ может содержать ссылку на скрипт решения (только git):



```
/usr/sbin/nologin
lylin
x
1000
1000
lylin,,,
/home/lylin
/bin/bash
systemd-coredump
x
999
999
systemd Core Dumper
/
/usr/sbin/nologin
user1
x
1500
1500

/home/user1
/bin/bash
user2
x
2000
2000
Пользователь 2,111,111-111,
/home/user2
/bin/bash
user3
x
2500
2500
user3,,,
/home/user3
/bin/bash
postgres
x
116
129
PostgreSQL administrator,,,
/var/lib/postgresql
/bin/bash
```



```

> echo $var
> done
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
_apt:x:103:65534::/nonexistent:/usr/sbin/nologin
astra-orientation:x:104:110::/var/cache/astra-orientation:/usr/sbin/nologin
messagebus:x:105:111::/nonexistent:/usr/sbin/nologin
pulse:x:106:115:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
fly-dm:x:107:118::/var/lib/fly-dm:/usr/sbin/nologin
sshd:x:108:65534::/run/ssh:/usr/sbin/nologin
ntp:x:109:121::/nonexistent:/usr/sbin/nologin
Debian-exim:x:110:122::/var/spool/exim4:/usr/sbin/nologin
dnsmasq:x:111:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
libvirt-qemu:x:64055:107:Libvirt Qemu,,,:/var/lib/libvirt:/usr/sbin/nologin
nm-openvpn:x:112:126:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
hplip:x:113:7:HPLIP system user,,,:/run/hplip:/bin/false
logcheck:x:114:127:logcheck system account,,,:/var/lib/logcheck:/usr/sbin/nologin
avahi:x:115:128:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
lylin:x:1000:1000:lylin,,,:/home/lylin:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
user1:x:1500:1500::/home/user1:/bin/bash
user2:x:2000:2000:Пользователь 2,111,111-111,,:/home/user2:/bin/bash
user3:x:2500:2500:user3,,,:/home/user3:/bin/bash
postgres:x:116:129:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
lylin@astra:~$

```

2.2. С помощью цикла создайте пользователей user4, user5 и user6. Отобразите в /etc/passwd только те строки, в которых есть слово user.

```

for users in user{4..6}
do
sudo useradd $users
done
cat /etc/passwd | grep user

```

student@prac-work-question:~# Предоставьте ответ в виде скриншота(-ов), где каждый шаг (действие) сопровождается письменным описанием.

Если необходимо предоставить скрипт, то ответ может содержать ссылку на скрипт решения (только git):



```

lylin@astra:~$ for users in user{4..6}
> do
> sudo useradd $users
> done
[sudo] пароль для lylin:
lylin@astra:~$ cat /etc/passwd | grep user
hplip:x:113:7:HPLIP system user,,,:/run/hplip:/bin/false
user1:x:1500:1500::/home/user1:/bin/bash
user2:x:2000:2000:Пользователь 2,111,111-111,:/home/user2:/bin/bash
user3:x:2500:2500:user3,,,:/home/user3:/bin/bash
user4:x:2501:2501::/home/user4:/bin/sh
user5:x:2502:2502::/home/user5:/bin/sh
user6:x:2503:2503::/home/user6:/bin/sh
lylin@astra:~$

```

3. Сценарий для crontab

3.1. Создайте сценарий `mem_monitor`, который выводит текущую дату и время, а также количество свободной оперативной памяти.

```

#!/bin/bash
echo -n `date` "  "
cat /proc/meminfo | grep MemFree

```

student@prac-work-question:~# Предоставьте ответ в виде скриншота(-ов), где каждый шаг (действие) сопровождается письменным описанием.

Если необходимо предоставить скрипт, то ответ может содержать ссылку на скрипт решения (только git):

```

lylin@astra:~$ #!/bin/bash
lylin@astra:~$ echo -n `date` "  "
Ср мар 26 22 07 32 MSK 2025 "  "lylin@astra:~$ cat /proc/meminfo | grep MemFree
MemFree:          1716612 kB

```

3.2. Создайте задание для `crontab`, которое будет выполнять сценарий `mem_monitor` и записывать результат в файл `memory`. Для более быстрой проверки результата сделайте так, чтобы сценарий отработывал каждую минуту.

```
crontab -e
```



```
* * * * * ~/mem_monitor >> ~/memory
```

student@prac-work-question:~# Предоставьте ответ в виде скриншота(-ов), где каждый шаг (действие) сопровождается письменным описанием.

Если необходимо предоставить скрипт, то ответ может содержать ссылку на скрипт решения (только git):

```
lylin@astra:~$ crontab -e
crontabs/lylin/: fdopen: Отказано в доступе
lylin@astra:~$ * * * * * ~/mem_monitor >> ~/memo
ry
bash: AstraLinux: команда не найдена
```



4. Дополнительно

4.1. Отобразите сетевые настройки интерфейса eth1. Используйте эти параметры для дальнейшей работы.

student@prac-work-question:~#
Предоставьте ответ в виде скриншота(-ов), где каждый шаг (действие) сопровождается письменным описанием.
Если необходимо предоставить скрипт, то ответ может содержать ссылку на скрипт решения (только git):

```
lylin@astra:~$ crontab -e
crontabs/lylin/: fdopen: Отказано в доступе
lylin@astra:~$ * * * * * ~/mem_monitor >> ~/memo
ry
bash: AstraLinux: команда не найдена
```

4.2. Напишите сценарий, который будет автоматически настраивать сетевые параметры для интерфейса eth1 с использованием статического адреса.

```
#!/bin/bash
con="Проводное соединение 2"
ip="A.B.C.D/M"
gw=" A.B.C.D/M"
dns=" A.B.C.D 8.8.8.8"

#Проверяем наличие соединения
if nmcli con show "$con" > /dev/null ; then
echo "Настраиваем «$con» на работу со статическим
адресом $ip gw $gw"

#Задаем адрес и адрес шлюза
nmcli con mod "$con" ip4.address $ip ipv4.gateway $gw

#Задаем адреса DNS
```



```

nmcli con mod "$con" ipv4.dns "$dns"

# Отключаем DHCP, переводим в "ручной" режим настройки
nmcli con mod "$con" ipv4.method manual
echo "Применены следующие настройки:"
nmcli -p con show "$con" | grep ipv4

# Перезапускаем соединение для применения новых
настроек
nmcli con down "$con" ; nmcli con up "$con"

else
echo "Соединение «$con» не найдено, настройте адрес
вручную"
exit 1
fi

```

student@prac-work-question:~# Предоставьте ответ в виде скриншота(-ов), где каждый шаг (действие) сопровождается письменным описанием.

Если необходимо предоставить скрипт, то ответ может содержать ссылку на скрипт решения (только git):

```

lylin@astra:~$ #!/bin/bash
lylin@astra:~$ con="Проводное соединение 2"
lylin@astra:~$ ip="A.B.C.D/M"
lylin@astra:~$ gw=" A.B.C.D/M"
lylin@astra:~$ dns=" A.B.C.D 8.8.8.8"
lylin@astra:~$ Проверяем наличие соединения
bash: Проверяем: команда не найдена
lylin@astra:~$ #Проверяем наличие соединения
lylin@astra:~$ if nmcli con show "$con" > /dev/null ; then
> echo "Настраиваем «$con» на работу со статическим адресом $ip gw $gw"
> #Задаем адрес и адрес шлюза
> nmcli con mod "$con" ipv4.address $ip ipv4.gateway $gw
> #Задаем адреса DNS
> nmcli con mod "$con" ipv4.dns "$dns"
> # Отключаем DHCP, переводим в "ручной" режим настройки
> nmcli con mod "$con" ipv4.method manual
> echo "Применены следующие настройки:"
> nmcli -p con show "$con" | grep ipv4
> # Перезапускаем соединение для применения новых настроек
> nmcli con down "$con" ; nmcli con up "$con"
> else
> echo "Соединение «$con» не найдено, настройте адрес вручную"
> exit 1
> fi

```



4.3. Для проверки отправьте эхо-запрос на адрес интерфейса eth1 и адрес шлюза.

student@prac-work-question:~#

Предоставьте ответ в виде скриншота(-ов), где каждый шаг (действие) сопровождается письменным описанием.

Если необходимо предоставить скрипт, то ответ может содержать ссылку на скрипт решения (только git):

```
lylin@astra:~$ #!/bin/bash
lylin@astra:~$ con="Проводное соединение 2"
lylin@astra:~$ ip="A.B.C.D/M"
lylin@astra:~$ gw=" A.B.C.D/M"
lylin@astra:~$ dns=" A.B.C.D 8.8.8.8"
lylin@astra:~$ Проверяем наличие соединения
bash: Проверяем: команда не найдена
lylin@astra:~$ #Проверяем наличие соединения
lylin@astra:~$ if nmcli con show "$con" > /dev/null ; then
> echo "Настраиваем «$con» на работу со статическим адресом $ip gw $gw"
> #Задаем адрес и адрес шлюза
> nmcli con mod "$con" ip4.address $ip ipv4.gateway $gw
> #Задаем адреса DNS
> nmcli con mod "$con" ipv4.dns "$dns"
> # Отключаем DHCP, переводим в "ручной" режим настройки
> nmcli con mod "$con" ipv4.method manual
> echo "Применены следующие настройки:"
> nmcli -p con show "$con" | grep ipv4
> # Перезапускаем соединение для применения новых настроек
> nmcli con down "$con" ; nmcli con up "$con"
> else
> echo "Соединение «$con» не найдено, настройте адрес вручную"
> exit 1
> fi
```

