

Лабораторная работа

Настройка оболочки.

1 Файлы инициализации.

При запуске в режиме сеансовой оболочки (при входе в систему) `bash` выполняет команды содержащиеся в файле `/etc/profile`, а затем ищет в домашнем каталоге пользователя файлы `.bash_profile`, `.bash_login` и `.profile` и выполняет команды содержащиеся в первом из найденных файлов. Если интерпретатор запускается в интерактивном режиме, но не в качестве сеансовой оболочки (например, при запуске `xterm`), то он выполняет команды содержащиеся в файле `.bashrc` из домашнего каталога пользователя. Наконец, если интерпретатор запускается не в интерактивном режиме (например, при выполнении сценария), то он, после запуска, выполняет команды содержащиеся в файле, имя которого хранится в переменной `BASH_ENV`.

Указанные файлы могут использоваться для настройки рабочей среды в соответствии с потребностями пользователя. Как правило в этих файлах содержатся присвоения значений переменным, определения алиасов и функций. Там же могут содержаться команды запуска программ.

2 Переменные используемые оболочкой

2.1 Переменные настройки локализации

Для определения языковых настроек используются следующие переменные окружения:

`LANG` определяет значение по умолчанию. Имеет самый низкий приоритет.

`LC_ALL`

если эта переменная определена, то ее значение перекрывает значения всех остальных переменных локализации.

`LC_COLLATE`

значение этой переменной определяет порядок сортировки символов.

`LC_CTYPE`

значение этой переменной указывает на используемые классы символов.

`LC_MESSAGES`

значение переменной определяет язык, на котором программы должны выводить сообщения.

`LC_NUMERIC`

определяет формат вывода чисел.

Этим переменным присваиваются значения вида:

`язык_СТРАНА.КОДОВАЯСТРАНИЦА`

или значение `C` для локали `POSIX`. Например, для использования русского языка следует присвоить переменной `LANG` значение `ru_RU.KOI8-R`.

Задание 1: Сравните вывод команд <code>date</code> и <code>LANG=C date</code>
--

2.2 Переменные используемые оболочкой и некоторыми программами

Следующие переменные влияют на работу оболочки и некоторых других программ:

- PATH** определяет путь поиска исполняемых файлов.
- HOME** домашний каталог пользователя.
- CDPATH**
путь поиска для команды `cd`.
- PROMPT_COMMAND**
перед выводом приглашения, интерпретатор выполняет команду, содержащуюся в данной переменной.
- PS1** значение этой переменной определяет вид приглашения командной строки. В данной переменной могут использоваться следующие специальные символы, предваренные обратной косой чертой:
- a** звуковой сигнал.
 - d** текущая дата.
 - e** escape символ.
 - h** короткое имя хоста.
 - H** длинное имя хоста.
 - l** имя терминала.
 - n** символ новой строки.
 - t** текущее время в 24-часовом формате.
 - T** текущее время в 12-часовом формате.
 - u** имя пользователя.
 - w** полный текущий рабочий каталог.
 - W** короткое имя текущего каталога.
 - \$** символ доллара если UID отличен от нуля и символ “#” если равен нулю.

Пример: `PS1="\d \T\n\u \$ "`

Задание 2: Измените приглашение командной строки так, чтобы в нем выводилось время, имя пользователя и текущий рабочий каталог. Заканчиваться приглашение должно символом доллара. Сохраните команду определяющую приглашение в файле `.bashrc`, чтобы приглашение восстанавливалось при каждом запуске оболочки.

- TMOUT** если значение данной переменной больше нуля, то оно определяет время, в секундах, в течение которого оболочка ожидает ввода пользователя.
- EDITOR**
многие программы используют значение данной переменной для определения предпочитаемого пользователем текстового редактора.
- PAGER** значение этой переменной используется, например, справочной системой `man` для определения программы просмотра справочных страниц.

2.3 Переменные устанавливаемые оболочкой

Следующие переменные устанавливаются самой оболочкой. Присваивать им значения в файлах инициализации не имеет смысла.

`PPID` номер родительского процесса оболочки.

`PWD` текущий рабочий каталог.

`OLDPWD`
каталог, который был рабочим до текущего.

`UID` идентификатор пользователя, запустившего оболочку.

`RANDOM`
при извлечении значения данной переменной возвращается случайное число от 0 до 32767.

`LINENO`
номер текущей строки сценария.

`HOSTNAME`
имя хоста.

Задание 3: Просмотрите значения переменных `PPID`, `PWD`, `OLDPWD`, `UID`, `RANDOM` и `HOSTNAME` и сохраните их в файле отчёта (`lab5.txt`) в формате “имя_переменной=значение”.

3 Псевдонимы команд

Bash позволяет определять псевдонимы для наиболее часто используемых команд. Например, если Вы часто используете команду `netstat -ran -f inet`, то, чтобы не набирать ее каждый раз, Вы можете определить для этой команды псевдоним `nsr` и использовать его вместо команды. Для определения такого псевдонима используется команда:

```
alias nsr='netstat -ran -f inet'
```

Псевдоним действует до завершения процесса оболочки. Если Вы хотите использовать псевдоним постоянно, то следует записать команду, определяющую псевдоним, в файл `.bashrc`.

Команда `alias` выполненная без параметров, выводит список определенных псевдонимов. Если в качестве параметра задано только имя, то выводится определение указанного псевдонима. Если команда выполнена в виде:

```
alias имя=команда
```

то создаётся новый (или переопределяется существующий) псевдоним. Первое слово команды может, в свою очередь, также являться псевдонимом, однако если оно совпадает с именем исходного псевдонима, то оболочка не будет его преобразовывать. Например, можно определить псевдоним `ls` для команды `ls -F`, и оболочка не будет раскрывать его рекурсивно. Для удаления псевдонима используется команда `unalias`. Псевдонимы можно использовать только в интерактивном режиме.

Задание 4: Создайте псевдонимы:

```
ls      для команды ls -G
ll      для команды ls -l
la      для команды ls -a
pa      для команды ps ax
x       для команды startx
```

Сохраните команды, определяющие псевдонимы в файле `.bashrc`, чтобы они выполнялись каждый раз при запуске оболочки.

4 Функции

Функции определяются следующим образом:

```
[ function ] имя() { список; }
```

Данная конструкция определяет функцию с указанным именем. Список команд, заключённый в фигурные скобки, является телом функции. Этот список выполняется каждый раз при вызове функции. Выполнение функции осуществляется в том же процессе, что и её вызов, т.е. все изменения окружения процесса остаются в силе после выхода из функции. Вызов функции осуществляется по её имени так же, как и запуск любой команды. Функции могут передаваться аргументы. Во время выполнения функции аргументы заменяют позиционные параметры (`$1`, `$2`, ...), после выхода из функции позиционные параметры восстанавливаются.

Команда `return` завершает выполнение функции и осуществляет переход к команде следующей за вызовом функции. Функции могут быть рекурсивными. Пример функции:

```
func1() { mkdir $1; touch $1/$2; }
```

Выполните вызов данной функции следующим образом:

```
func1 dir1 file1
```

Задание 5: Определите функцию `lab`, которая будет выводить строку "Лабораторная работа N", где N передаётся функции в качестве аргумента, а затем строку содержащую Ваши имя и фамилию. Сохраните определение функции в файле `.bashrc`. Добавьте результат работы функции в файл отчета.