

Лабораторная работа № U6

Возможности интерпретатора командной строки

Интерпретатор командной строки (shell) обладает всеми свойствами языка программирования – переменные, логические условия, циклы.

Переменные

Указать значение переменной можно varname=value

Просмотреть уже определённые переменные SHELL можно с помощью команды env (выводит все переменные окружения) или с помощью, например, echo \$VARNAME, где VARNAME – это название переменной, значение которой надо вывести.

Предопределённые системные переменные PATH, PS1, HOME, MAIL и д.р.

Определение идентификатора текущего процесса - \$\$

Назначить переменную VARNAME – переменной окружения – export VARNAME

Удалить переменную из памяти - unset

Некоторые команды для работы с переменными

- команда echo выводит в выходной поток значения своих аргументов;
- команда expr выполняет арифметические действия над своими аргументами;
- команда eval обеспечивает дополнительный уровень подстановки своих аргументов, а затем их выполнение;
- команда test с соответствующими ключами проверяет необходимое условие;
- команда sleep служит для реализации задержки.

При обработке переменных есть три варианта кавычек:

echo "ls \$var" - при обработке команды переменные будут «раскрыты»

echo 'ls \$var' - печатает как есть

echo `ls \$var` - перед печатью текста происходит исполнение команды, указанной в кавычках

Работа с аргументами/опциями указываемыми при запуске

➤ при написании скрипта:

1. \$# - число аргументов, переданных в Shell;
2. \$0, \$1, \$2 ... - параметры переданные в Shell

➤ При написании программ на C:

```
int main(int argc, char **argv)
```

или

```
int main(int argc, char **argv, char **env)
```

где argc — число параметров, argv — массив указателей на строки параметров командной строки, env — массив указателей на строки переменных окружения. Каждая строка кодирована в формате ИМЯ=ЗНАЧЕНИЕ

Логические условия

```
if условие then
```

```
....
```

```
else
```

```
.....
```

```
fi
```

Проверить логическое условие – test или [], например [-f /filename] проверяет наличие файла filename в корневом каталоге , или для проверки что указанное имя является каталогом -d.

Также есть *case*, аналогичная по функциям *switch* языка C

Операторы циклов

```
for _переменная_цикла_ in _где_искать_; do  
...операторы цикла;  
done
```

```
for i in `ls`; do mv "$i" `echo "$i" | tr "[:upper:]" "[:lower:]"`; done
```

Задание

- Разработать скрипт на shell, который будет проверять содержимое каталога, указанного в первом входном параметре (контроль входных параметров НЕ нужно реализовывать) и выводить листинг в примерном виде
Имя1 -> файл
Имя2 -> файл
Имя3 -> директория
Имя4 -> файл