

# Лабораторная работа №11

Выполнил Бабков Дмитрий Николаевич, НПМбд-01-20

# Цель работы

Изучить основы программирования в оболочке ОС UNIX/Linux. Научиться писать небольшие командные файлы. # Задание Написать командные файлы для выполнения заданий

# Задание

Написать командные файлы для выполнения заданий

# Ход работы

## Первое задание

Скрипт, делающий бэкап самого себя в директорию `~/backup` (Рис.1)

```
task1.sh      [----] 36 L:[ 1+ 1  2/ 2] *(144 / 282b) 0009 0x009
backup=~/backup><-----><-----><----->#Переменной присваивается путь в директорию backup
tar -zcvf ${backup}/back.gz task1.sh<-->#Выполняется копирование с архивированием этого набора команд в архив back.gz
```

## Второе задание

Командный файл, обрабатывающий любое, кол-во аргументов командной строки и выводящий их на экран (Рис.2)

```
task2.sh      [-M--] 41 L:[ 1+ 2 3/ 4] *(143 / 148b) 0010 0x00A
for param in "$@"<----->#Цикл, выполняющийся пока есть параметры
do
echo "$param"<-><----->#Вывод параметров
done
```

# Третье задание

Командный файл, являющийся аналогом команды ls. Выводит имена всех файлов заданной директории и их разрешения (Рис.3)

```
task3Mod.sh      [----] 0 L: [ 1+ 0  1/ 30] *(0  /1298b) 0112 0x070
print_perm() {<--<-----<----->#Функция для определения разрешений файла
    case "$1" in
        0) printf "NO PERMISSIONS";;
        1) printf "Execute only";;
        2) printf "Write only";;
        3) printf "Write & execute";;
        4) printf "Read only";;
        5) printf "Read & execute";;
        6) printf "Read & write";;
        7) printf "Read & write & execute";;
    esac
}

path=$1<-----<-----<----->#Переменной path присваивается значение первого оператора -.

for f in $path/*; <-----<----->#Проходятся все файлы, расположенные в директории, в которую ведет путь
do
    perm=$(stat -c%a "$f")<----->#Передается информация о разрешениях
    user=${perm:0:1}<--<----->#Разрешения пользователя
    group=${perm:1:1}<--<----->#Группы
    global=${perm:2:1}<--<----->#Других

    b=$(basename $f)<-->#Переменной b присваивается имя файла (без пути)
    echo $b<-----<----->#Выводится имя файла
    printf "\tOwner Access: $(print_perm $user)\n"<----->#Вызывается ранее описанная функция
    printf "\tGroup Access: $(print_perm $group)\n"
    printf "\tothers Access: $(print_perm $global)\n"
    echo ""
done
```

## Четвертое задание

Командный файл, получающий в качестве аргументов имя директории и расширение. Находит кол-во файлов с заданным расширением (Рис.4)

```
task4.sh      [-M--] 67 L:[ 1+ 5  6/ 11] *(469 / 717b) 0010 0x00A
path=$1<-----><----->#Присвоение переменной первого параметра (путь)
ext=$2<-----><----->#Присвоение переменной второго параметра (расширение)
count=0<-----><----->#Счетчик
for f in $path/*$ext<-->#Цикл, идущий по всем файлам, в директории path с расширением ext
do
    b=$(basename $f)<-->#Присвоение переменной имени файла без пути
    echo $b<-----><----->#Вывод файла с заданным расширением
    ((count++))<----->#Инкрементирование счетчика
done
echo $count<-----><----->#Вывод кол-ва файлов с заданным расширением
```

## Вывод

В ходе выполнения лабораторной работы я изучил основы программирования в оболочке ОС UNIX/Linux и научился писать небольшие командные файлы

**Спасибо за внимание**