

Федеральное агентство связи
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Сибирский государственный университет
телекоммуникаций и информатики»

Лабораторная работа №1
«Часть 2 – Создание простых скриптов»

Выполнил: студент 4 курса

ИВТ, гр. ИП-713

Михеев Н.А.

Проверил: старший преподаватель кафедры ПМиК

Милешко А.В.

Новосибирск, 2020 г.

1. Написать скрипт, который просто выводит значения переданных ему параметров.

Для написания данного скрипта был предварительно создан файл “lab1_2_1”, так же ему были даны права для выполнения командой `# chmod u+x lab1_2_1`.

Далее был написан сам скрипт, используя в качестве интерпретатора Korn:

```
#!/bin/sh
for i in "$@"; do
    echo $1
done
```

```
# ./lab1_2_1
# ./lab1_2_1 arg1
arg1
# ./lab1_2_1 arg1 arg2 arg3 arg4 arg5 qwerty
arg1
arg2
arg3
arg4
arg5
qwerty
#
```

Результат работы скрипта с 0 переданных аргументов, с 1 аргументом и с 6 различными аргументами, как видно скрипт работает корректно.

2. Написать скрипт, который с помощью утилит *pidin* и *grep* выводит на экран информацию об указанном по имени процессе.

Для реализации задания был написан скрипт:

```
#!/bin/sh
pidin | grep $1
```

Данный скрипт требует только один аргумент – имя процесса. Получив имя процесса запускается сам скрипт, внутри которого pipeline из команд `pidin` и `grep` результата по переданному имени процесса через аргумент.

```
# ./lab1_2_1 bin/pidin
946214  1 bin/pidin          10r  REPLY          1
# ./lab1_2_1 bin/login
188437  1 bin/login             10o  REPLY          4103
188438  1 bin/login             10o  REPLY          4103
188440  1 bin/login             10o  REPLY          4103
266265  1 bin/login             10o  REPLY          4103
#
```

Результат работы скрипта: передаем один параметр – сам процесс `bin/pidin`, затем `bin/login` – данных процессов несколько, соответственно результат многострочный.

3. Написать скрипт, который компилирует указанную программу и при отсутствии ошибок запускает её. Если же есть ошибки, то автоматически вызывает редактор для их исправления.

Перед написанием скрипта была разработана программа test.c на языке C, в которой происходит вывод строки: “Hello, World!”. Далее уже был разработан сам скрипт:

```
#!/bin/sh
cc test.c
if [ -f a.out ]; then
    ./a.out
else
    sleep 5
    vi test.c
fi
```

Данный скрипт сначала компилирует указанную программу, далее следует проверка – был ли создан исполняемый файл, если да – запускает его; если нет – то компилятор выведет ошибку, есть 5 секунд чтобы ее прочитать и затем запускается редактор vi для исправления ошибок.

Пример удачной компиляции и работы скрипта:

```
# ./lab1_2_1
Hello, World!
# _
```

Пример неудачной работы скрипта:

```
# ./lab1_2_1
test.o: In function `main':
test.c:(.text+0x11): undefined reference to `printf'
cc: /usr/qnx650/host/qnx6/x86/usr/bin/ntox86-ld error 1
skipping 4 old session files
reading test.c
_
```

Вывод ошибок компиляции

