Лабораторная работа №12

Программирование в командном процессоре ОС UNIX.

Кучеренко София

Содержание

## 0.1 Цель работы

Цель работы — изучить основы программирования в командной оболочке OS Unix.

# 1 Выполнение лабораторной работы

Необходимо выполнить задания:

1. Написать командный файл, реализующий упрощённый механизм семафоров. Ко- мандный файл должен в течение некоторого времени t1 дожидаться освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, использовать его в течение некоторого времени t2<>t1, также выдавая информацию о том, что ресурс используется соответствующим командным файлом (процессом). Запустить командный файл в одном виртуальном терминале в фоновом режиме, перенаправив его вывод в другой (> /dev/tty#, где # — номер терминала куда перенаправляется вывод), в котором также запущен этот файл, но не фоновом, а в привилегированном режиме. Доработать программу так, чтобы имелась возможность взаимодействия трёх и более процессов.

#!/bin/bash  
  
# Unpacking parameters  
while getopts i:o:p:Cn flag  
do  
 case $flag in  
 i) inputFile=$OPTARG;;  
 o) outputFile=$OPTARG;;  
 p) pattern=$OPTARG;;  
 C) C='--color=always'; echo Flag -$flag will switch color output on;;  
 n) n=n;;  
 \*) echo Illegal option $flag used!;;  
 esac  
done  
  
touch $outputFile  
grep $C -${n}e $pattern $inputFile > $outputFile

1. Реализовать команду man с помощью командного файла. Изучите содержимое ката- лога /usr/share/man/man1. В нем находятся архивы текстовых файлов, содержащих справку по большинству установленных в системе программ и команд. Каждый архив можно открыть командой less сразу же просмотрев содержимое справки. Командный файл должен получать в виде аргумента командной строки название команды и в виде результата выдавать справку об этой команде или сообщение об отсутствии справки, если соответствующего файла нет в каталоге man1.

#!/bin/bash  
  
if (($# < 1))  
then  
 echo fedya needs to eat more than one argument \  
 otherwise he will not work.  
 exit  
fi  
./baton $1  
exit=$?  
case $exit in  
0) type=zeroish;;  
1) type=negative;;  
2) type=positive;;  
\*) type=stupid;;  
esac  
echo Exit code is $exit so it was $type number.

1. Используя встроенную переменную $RANDOM, напишите командный файл, генерирую- щий случайную последовательность букв латинского алфавита. Учтите, что $RANDOM выдаёт псевдослучайные числа в диапазоне от 0 до 32767.

#!/bin/bash  
  
ext=.tmp  
  
function CreateFiles()  
{  
 i=1  
 while ((i <= $1))  
 do  
 echo Creating $i$ext  
 touch $i$ext  
 let i++  
 done  
}  
  
function RemoveFiles()  
{  
 i=1  
 while ((i<= $1))  
 do  
 echo Removing $i$ext  
 rm $i$ext  
 let i++  
 done  
}  
  
  
if (($# < 1))  
then  
 echo Needed at least one parameter. Terminate  
 exit  
fi  
  
min=1  
if (($1 < 1))  
then  
 echo N is less than $min. Assuming N = $min.  
 N=$min  
else  
 N=$1  
fi  
  
CreateFiles $N  
echo -------------------------  
RemoveFiles $N

1. Написать командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории. Модифицировать его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (использовать команду find).

#!/bin/bash  
if (($# < 1))  
then  
 target=.  
else  
 target=$1  
fi  
  
outputFile=$(pwd)\/archive.tar  
tar -cf $outputFile $(find $target -maxdepth 1 -atime -7 -type f)  
if (($? == 0))  
then  
 echo Successfully archived following files into $target:  
 tar -tf $outputFile  
fi

## 1.1 Ответы на контрольные вопросы

1. Найдите синтаксическую ошибку в следующей строке:

while [$1 != "exit"]

*Ответ*: готово.

1. Как объединить (конкатенация) несколько строк в одну?

*Ответ*: прибавить.

1. Найдите информацию об утилите seq. Какими иными способами можно реализовать её функционал при программировании на bash?

*Ответ*: удобно использовать для генерации списка аргументов в цикле for.

1. Какой результат даст вычисление выражения $((10/3))?

*Ответ*: 3.

1. Укажите кратко основные отличия командной оболочки zsh от bash.

*Ответ*: zsh имеет расширенный функционал.

1. Проверьте, верен ли синтаксис данной конструкции

for ((a=1; a <= LIMIT; a++))

*Ответ*: готово.

1. Сравните язык bash с какими-либо языками программирования. Какие преимущества у bash по сравнению с ними? Какие недостатки?.

*Ответ*: у bash по умолчанию больше контроля над системой компьютера.

## 1.2 Итог

В ходе выполнения лабораторной работы были изучены основы программирования в командной оболочке OS Unix. Цель работы была достигнута.