

Front matter

lang: ru-RU title: "Отчет по лабораторной работе №13" subtitle: "по дисциплине: Операционные системы" author: "Трефилова Мария Андреевна"

Formatting

toc-title: "Содержание" toc: true # Table of contents toc_depth: 2 lof: false # List of figures lot: false # List of tables fontsize: 12pt linestretch: 1.5 papersize: a4paper documentclass: scrreprt polyglossia-lang: russian polyglossia-otherlangs: english mainfont: PT Serif romanfont: PT Serif sansfont: PT Sans monofont: PT Mono mainfontoptions: Ligatures=TeX romanfontoptions: Ligatures=TeX sansfontoptions: Ligatures=TeX,Scale=MatchLowercase monofontoptions: Scale=MatchLowercase indent: true pdf-engine: lualatex header-includes: - \linepenalty=10 # the penalty added to the badness of each line within a paragraph (no associated penalty node) Increasing the value makes tex try to have fewer lines in the paragraph. - \interlinepenalty=0 # value of the penalty (node) added after each line of a paragraph. - \hyphenpenalty=50 # the penalty for line breaking at an automatically inserted hyphen - \exhyphenpenalty=50 # the penalty for line breaking at an explicit hyphen - \binoppenalty=700 # the penalty for breaking a line at a binary operator - \relpenalty=500 # the penalty for breaking a line at a relation - \clubpenalty=150 # extra penalty for breaking after first line of a paragraph - \widowpenalty=150 # extra penalty for breaking before last line of a paragraph - \displaywidowpenalty=50 # extra penalty for breaking before last line before a display math - \brokenpenalty=100 # extra penalty for page breaking after a hyphenated line - \predisplaypenalty=10000 # penalty for breaking before a display - \postdisplaypenalty=0 # penalty for breaking after a display - \floatingpenalty = 20000 # penalty for splitting an insertion (can only be split footnote in standard LaTeX) - \raggedbottom # or \flushbottom - \usepackage{float} # keep figures where there are in the text

- \floatplacement{figure}{H} # keep figures where there are in the text

Цель работы

Изучить основы программирования в оболочке ОС UNIX. Научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Выполнение лабораторной работы

Начальный этап

1. Напишем командный файл, реализующий упрощённый механизм семафоров. Командный 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)

- : mcedit — Konsole

ФайлПравкаВидЗакладкиНастройкаСправка

lab10_1.sh [B---] 0 L:[1* 0 1/ 18] *(0 / 231b) 0108 0x06C

lockfile="./locking.file"

exec {fn}>\$lockfile

if test -f "\$lockFile"

then

while [!!=0]

do

if flock -n \${fn}

then

echo "File was locked"

sleep 2

echo "Unlocking..."

flock -u \${fn}

else

echo "File already locked"

sleep 2

fi

done

fi

[illegible]

2. Реализуем команду `map` с помощью командного файла.

```
lab10_2.sh [BM--] 0 L: 1* 0 1/ 10] *(0 / 179b) 0099 0x063
command=""
while getopts "n" opt
do
case $opt in
n) command="$OPTARG";;
esac
done
if test -f "/usr/share/man/man1/$command/.1.gz"
then less /usr/share/man/man1/$command.1.gz
fi
```

```
matrefilova@edk4n70 ~ $ ncedit lab10_2.sh
matrefilova@edk4n70 ~ $ chmod ugo+rxw lab10_2.sh
matrefilova@edk4n70 ~ $ ./lab10_2.sh
matrefilova@edk4n70 ~ $ ./lab10_2.sh -n rm
```

3. Используя встроенную переменную \$RANDOM, напомним командный файл, генерирующий случайную последовательность букв латинского алфавита

```
echo "My password generator"
cat /dev/urandom | tr -dc "a-zA-Z0-9" | fold -w 10 | head -n 1
echo "Prog by Your tech requirement: "
echo $RANDOM | tr '0-9' 'a-zA-Z'
```

```
matrefilova@dk4n70 ~ $ ./lab10_3.sh
My password generator
ilHqeFSABk
Prog by Your tech requirement:
cffde
matrefilova@dk4n70 ~ $ ./lab10_3.sh
My password generator
VVGNgMnhOk
Prog by Your tech requirement:
bgdjg
matrefilova@dk4n70 ~ $ ./lab10_3.sh
My password generator
CfMcixezws
Prog by Your tech requirement:
daggi
matrefilova@dk4n70 ~ $ ./lab10_3.sh
My password generator
fZoE73YAcH
Prog by Your tech requirement:
hcia
```

Выводы

Таким образом, я изучила основы программирования в оболочке ОС UNIX. Научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Контрольные вопросы

1. Найдите синтаксическую ошибку в следующей строке: while [\$1 != "exit"] \$1. Так же между скобками должны быть пробелы. В противном случае скобки и рядом стоящие символы будут восприниматься как одно целое
2. Как объединить (конкатенация) несколько строк в одну? cat file.txt | xargs | sed -e 's/. /\n/g'
3. Найдите информацию об утилите seq. Какими иными способами можно реализовать её функционал при программировании на bash? seq - выдает последовательность чисел. Реализовать ее функционал можно командой for n in {1..5} do <КОМАНДА> done
4. Какой результат даст вычисление выражения $\$(10/3)?$ 3
5. Укажите кратко основные отличия командной оболочки zsh от bash. Zsh очень сильно упрощает работу. Но существуют различия. Например, в zsh после for обязательно вставлять пробел, нумерация массивов в zsh начинается с 1 (что не особо удобно на самом деле). Если вы собираетесь писать скрипт, который легко будет запускать множество разработчиков, то я рекомендую Bash. Если скрипты вам не нужны - Zsh (более простая работа с файлами, например)
6. Проверьте, верен ли синтаксис данной конструкции for ((a=1; a <= LIMIT; a++)) Верен
7. Сравните язык bash с какими-либо языками программирования. Какие преимущества у bash по сравнению с ними? Какие недостатки? Bash позволяет очень легко работать с файловой системой без лишних конструкций (в отличие от обычного языка программирования). Но относительно обычных языков программирования bash очень сжат. Тот же Си имеет гораздо более широкие возможности для разработчика. **Вывод:** В ходе лабораторной работы я изучил основы программирования в оболочке ОС UNIX. Научился писать более сложные командные файлы с использованием логических управляющих конструкций и циклов