

Отчёт по лабораторной работе №14

Дисциплина: Архитектура компьютера и операционные системы

Ванюшкина Татьяна Валерьевна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	8

Список иллюстраций

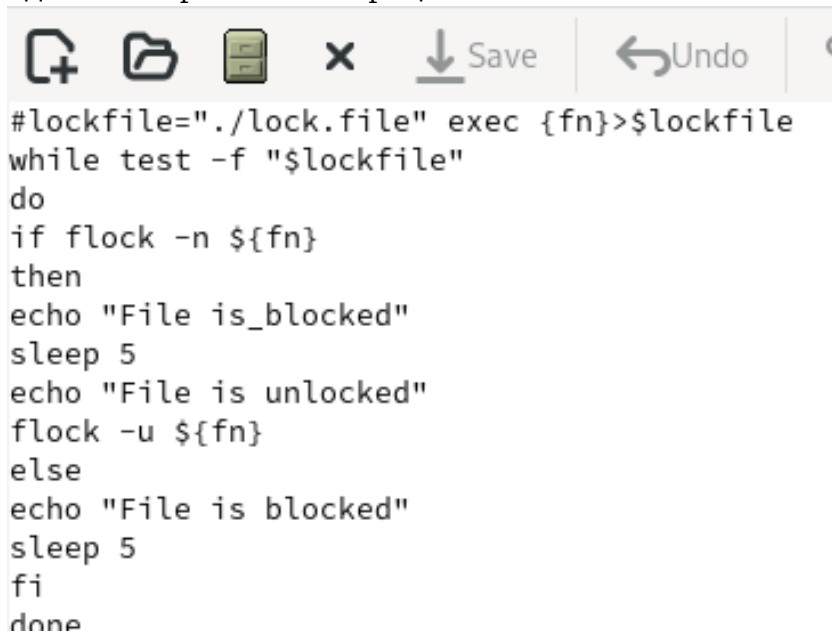
Список таблиц

1 Цель работы

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

2 Выполнение лабораторной работы

Пишу командный файл, реализующий упрощённый механизм семафоров. Командный файл должен в течение некоторого времени t_1 дожидается освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения, использовать его в течение некоторого времени $t_2 < t_1$, также выдавая информацию о том, что ресурс используется соответствующим командным файлом (процессом). Запускаю 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 имела возможность взаимодействия трёх и более процессов.



```
#lockfile="./lock.file" exec {fn}>$lockfile
while test -f "$lockfile"
do
if flock -n ${fn}
then
echo "File is_blocked"
sleep 5
echo "File is unlocked"
flock -u ${fn}
else
echo "File is blocked"
sleep 5
fi
done
```

(рис.1 ??)

{#fig:001}

Реализую команду `map` с помощью командного файла. Изучаю 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.

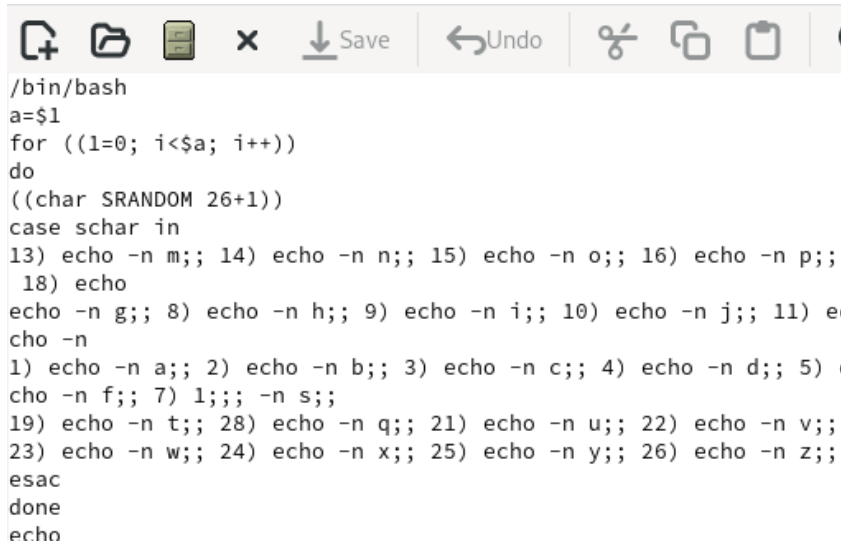


```
#!/bin/bash
a=$1
if test -f "/usr/share/man/man1/$a.1.gz"
then less /usr/share/man/man1/$a.1.gz
else
echo "There is no such command"
fi
```

(рис.2 ??)

{#fig:002}

Пишу командный файл, генерирующий случайную последовательность букв латинского алфавита. \$RANDOM выдаёт псевдослучайные числа в диапазоне от 0 до 32767.



```
/bin/bash
a=$1
for ((l=0; l<$a; l++))
do
((char=$((RANDOM%26+1)))
case char in
13) echo -n m;; 14) echo -n n;; 15) echo -n o;; 16) echo -n p;;
18) echo
echo -n g;; 8) echo -n h;; 9) echo -n i;; 10) echo -n j;; 11) echo -n
1) echo -n a;; 2) echo -n b;; 3) echo -n c;; 4) echo -n d;; 5) echo -n e;;
6) echo -n f;; 7) echo -n l;; 20) echo -n s;;
19) echo -n t;; 28) echo -n q;; 21) echo -n u;; 22) echo -n v;;
23) echo -n w;; 24) echo -n x;; 25) echo -n y;; 26) echo -n z;;
esac
done
echo
```

(рис.3 ??)

{#fig:003}

3 Выводы

В ходе выполнения лабораторной работы я изучила основы программирования в оболочке ОС UNIX и научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов # Список литературы{.unnumbered}

Курс: Архитектура компьютеров и операционные системы. Раздел “Операционные системы” (02.03.00, УГСН) (rudn.ru)