Задание 1. Создайте скрипт, который предоставляет пользователю

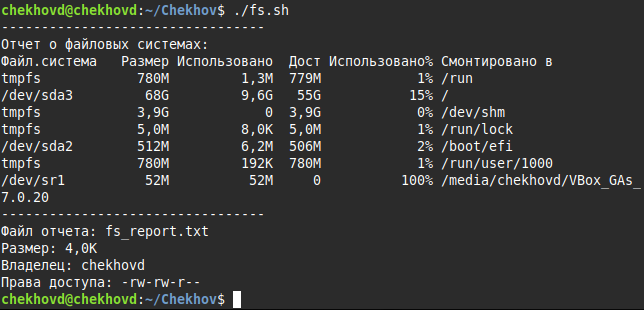
следующие функции:

1. Создает отчет о количестве доступных файловых систем, их

свободном и занятом пространстве;

2. Выводит информацию об имени и размере файла отчета, а также о

правах пользователя на него.



#!/bin/bash

REPORT="fs\_report.txt"

echo "Отчет о файловых системах:" > "$REPORT"

df -h >> "$REPORT"

echo "---------------------------------"

cat "$REPORT"

echo "---------------------------------"

echo "Файл отчета: $REPORT"

echo "Размер: $(du -h "$REPORT" | awk '{print $1}')"

echo "Владелец: $(ls -l "$REPORT" | awk '{print $3}')"

echo "Права доступа: $(ls -l "$REPORT" | awk '{print $1}')"

Задание 2. Создайте скрипт, который предоставляет пользователю

следующие функции:

1. Создает отчет об именах файлов их инодах, правах, размерах и

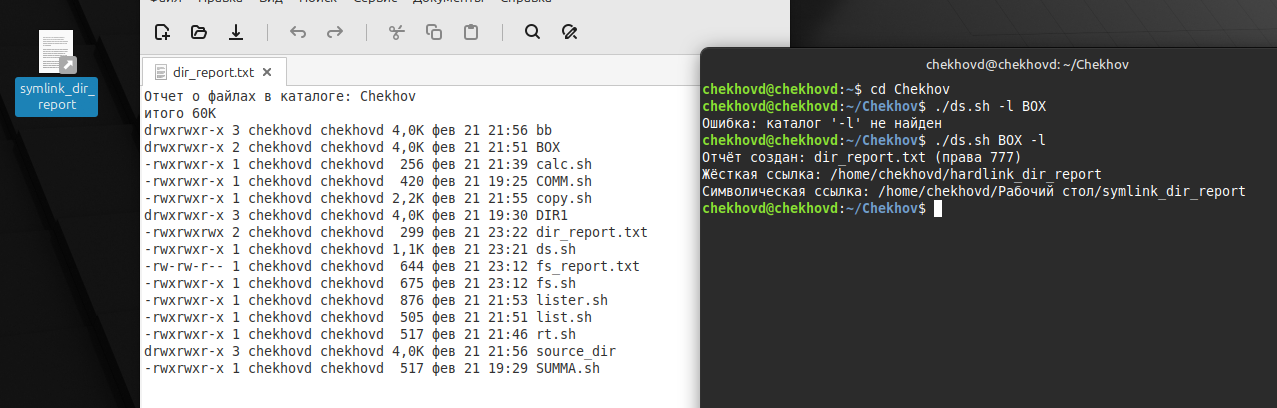
временах последнего доступа для каталога, указанного пользователем (отчет

формируется в порядке возрастания инодов);

2. Открывает полные права на файл отчета для всех пользователей;

3. Создает жесткую ссылку на файл отчета в домашнем каталоге

пользователя и символьную ссылку на рабочем столе.



#!/bin/bash

if [ -z "$1" ]; then

echo "Ошибка: укажите каталог"

exit 1

fi

DIR="$1"

REPORT="dir\_report.txt"

HARDLINK="$HOME/hardlink\_dir\_report"

DESKTOP="$HOME/Рабочий стол"

SYMLINK="$DESKTOP/symlink\_dir\_report"

if [ ! -d "$DIR" ]; then

echo "Ошибка: каталог '$DIR' не найден"

exit 1

fi

# -i показывает номер inode

# -l длинный формат (права, владелец и т.д.)

# -h человекочитаемые размеры

# --time=atime показываем время последнего доступа

echo "Отчёт о файлах в каталоге: $DIR" > "$REPORT"

ls -lih --time=atime "$DIR" >> "$REPORT"

chmod 777 "$REPORT"

[ -e "$HARDLINK" ] && rm -f "$HARDLINK"

[ -e "$SYMLINK" ] && rm -f "$SYMLINK"

ln "$REPORT" "$HARDLINK"

if [ -d "$DESKTOP" ]; then

ln -s "$(realpath "$REPORT")" "$SYMLINK"

else

echo "Внимание: Каталог '$DESKTOP' не найден, символьная ссылка не создана!"

fi

echo "Отчёт создан: $REPORT "

echo "Жёсткая ссылка: $HARDLINK"

if [ -d "$DESKTOP" ]; then

echo "Символическая ссылка: $SYMLINK"

fi

Задание 3. Создайте скрипт, который предоставляет пользователю

следующие возможности:

1. Выясняет, зарегистрирован ли в системе пользователь с заданным

именем.

2. Выясняет, работает ли в текущий момент пользователь с заданным

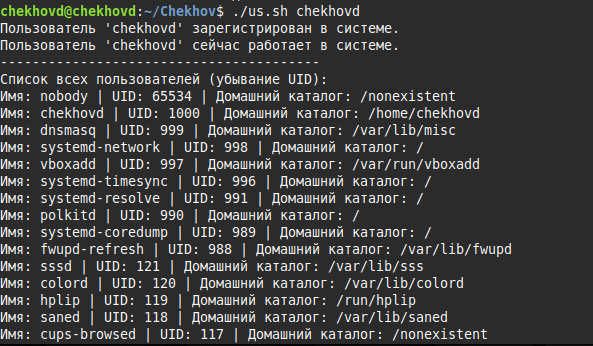
именем.

3. Выводит список зарегистрированных пользователей с именами их

домашних каталогов и числовыми идентификаторами, в порядке убывания

идентификаторов (для редактирования вывода используйте команды grep

или sed).



#!/bin/bash

if [ -z "$1" ]; then

echo "Использование: $0 <имя\_пользователя>"

exit 1

fi

USER\_NAME="$1"

if id "$USER\_NAME" &>/dev/null; then

echo "Пользователь '$USER\_NAME' зарегистрирован в системе."

else

echo "Пользователь '$USER\_NAME' не найден в системе."

exit 0

fi

if who | grep -wq "$USER\_NAME"; then

echo "Пользователь '$USER\_NAME' сейчас работает в системе."

else

echo "Пользователь '$USER\_NAME' сейчас не в системе."

fi

echo "----------------------------------------"

echo "Список всех пользователей (убывание UID):"

cut -d: -f1,3,6 /etc/passwd | sort -t: -k2 -nr | \

sed 's/\([^:]\*\):\([^:]\*\):\(.\*\)/Имя: \1 | UID: \2 | Домашний каталог: \3/'

Задание 4. Реализуйте два исполняемых файла, выполняющих

следующие задачи:

1. Расчет значения функции sin(x) с шагом 0.01 от 0 до бесконечности.

2. Расчет значения функции cos(x) с шагом 0.001 от 0 до бесконечности.

3. Для реализации исполняемых файлов можно воспользоваться

компиляторами gcc и g++.

4. Запустите первый файл в фоновом режиме.

5. Запустите второй файл в приоритетном режиме.

6. Переведите второй процесс в фоновый режим.

7. Переведите первый процесс в приоритетный режим.

8. Для каждого из процессов выведите:

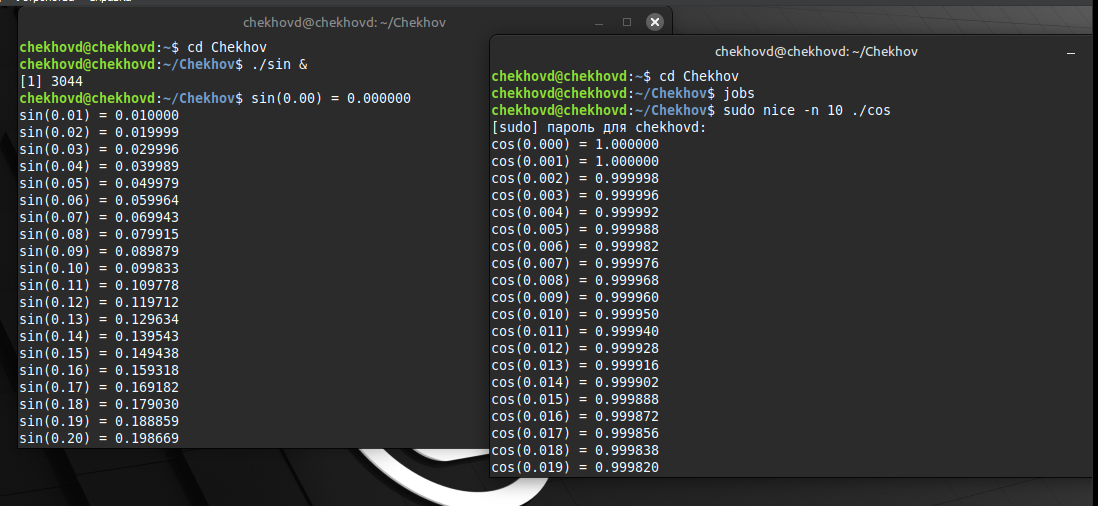
- долю оперативной памяти, занимаемой процессом

- время, прошедшее с момента запуска процесса

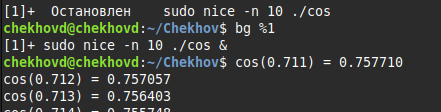
- имя управляющего терминала

- время старта процесса

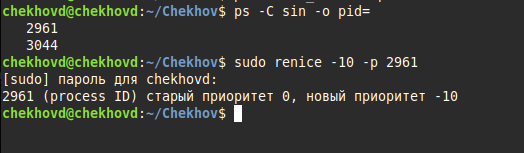
4,5



6



7



8

