МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра Вычислительной техники

ОТЧЁТ

по лабораторной работе №1

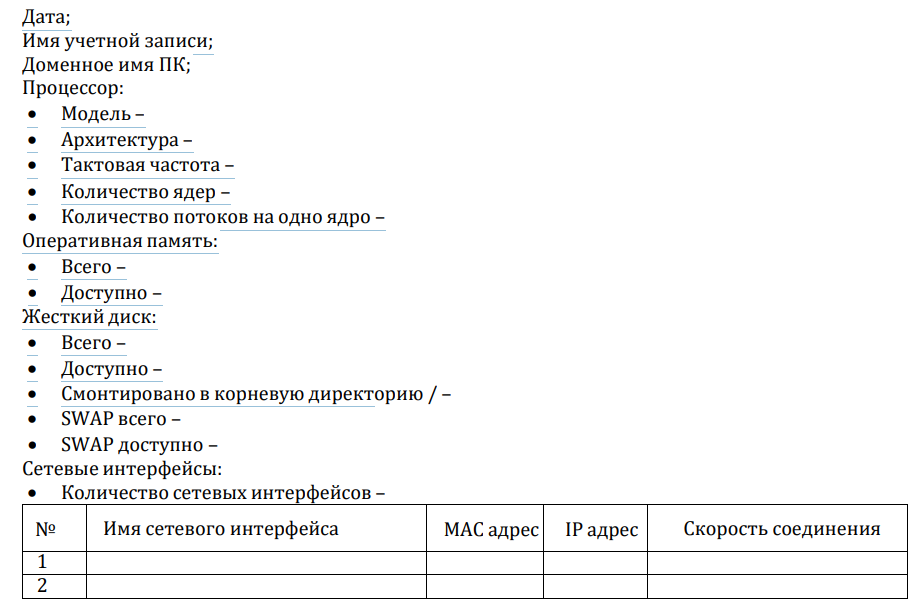
«Оценка характеристик персонального компьютера»

по дисциплине: «Системы параллельной обработки данных»

|  |  |
| --- | --- |
| Выполнил:  Студент гр. АПИМ-25, АВТФ:  Клименко К. В. | Преподаватель:  Перышкова Е. Н. |

Новосибирск, 2025

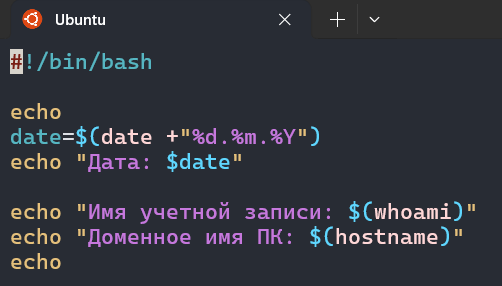
**Задание:** написать bash-скрипт, который выводит на экран характеристики ПК в следующем формате:



**Ход работы:**

Первоначально для выполнения задания был создан файл скрипта lab1script.sh. Само написание скрипта производилось с помощью утилиты vim. Для выведения текущей даты была использована команда date с форматированием “%d.%m.%Y”. Для выведения имени учетной записи была использована команда whoami, а для доменного имени ПК – hostname.

Рисунок 1. Дата, имя учетной записи и доменное имя ПК

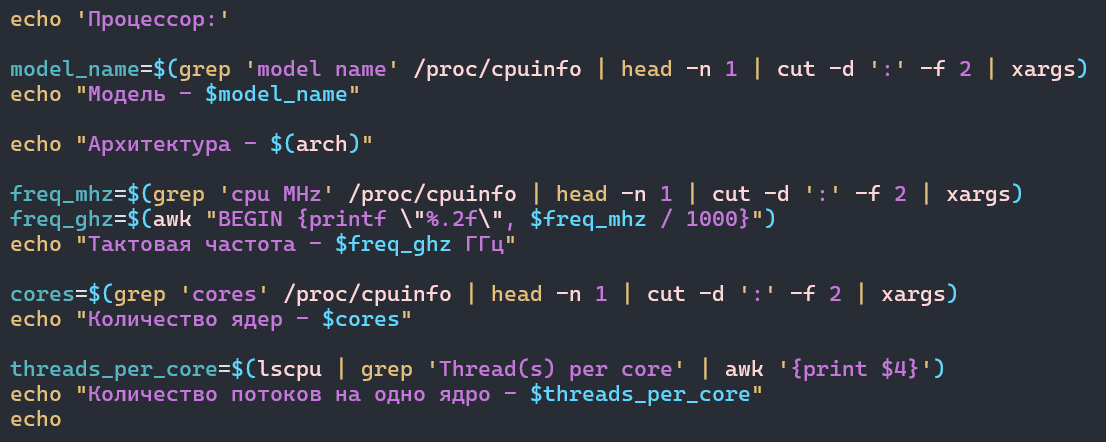


Далее, для выведения имени модели процессора с помощью команды grep произвелся парсинг файла /proc/cpuinfo, содержащий информацию о процессоре, на строку «model name», далее через пайплайн вывод обрезался до первой строки с помощью команды head, помощью команды cut вывод разделился по разделителю «:» и взято второе поле, содержащее непосредственно имя модели. Команда xargs обрезала пробелы в начале и конце строки и нормализовала пробелы внутри. Для вывода архитектуры процессора была использована команда arch.

Для того чтобы получить тактовую частоту аналогичным с моделью образом, произвелся парсинг /proc/cpuinfo с помощью команды grep, а затем с помощью встроенного в bash языка программирования awk в блоке BEGIN полученная ранее частота в мегагерцах была переведена в гигагерцы и округлена до сотых. Количество ядер процессора было получено таким же образом, как и модель.

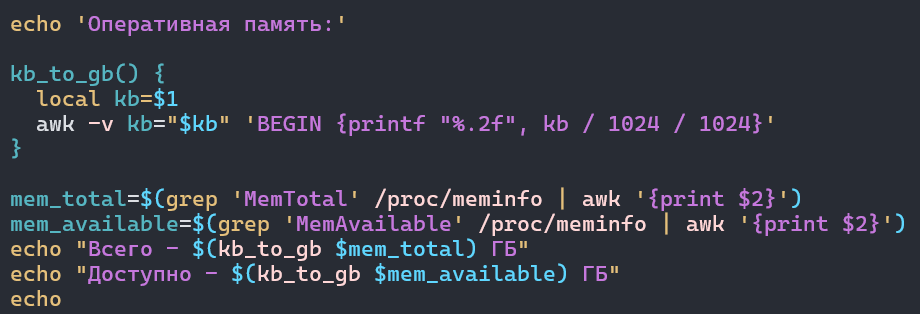
Для получения количества потоков на одно ядро, была задействована команда lscpu, выводящая данные о процессоре на основе /proc/cpuinfo, вывод которой был пропарсен на ключевую строку, а awk, функция print по умолчанию которого делит строку по пробелу, вывел 4 поле (токен), содержащий непосредственно количество потоков на одно ядро процессора.

Рисунок 2. Процессор



Перед выводом количества оперативной памяти, была создана вспомогательная функция kb\_to\_gb, посредством языка awk переводящая килобайты в гигабайты и форматирующая вывод. Для получения общего и доступного количества RAM, через grep произвелся парсинг файла /proc/meminfo, а функция kb\_to\_gb перевела полученные значения из килобайтов в гигабайты.

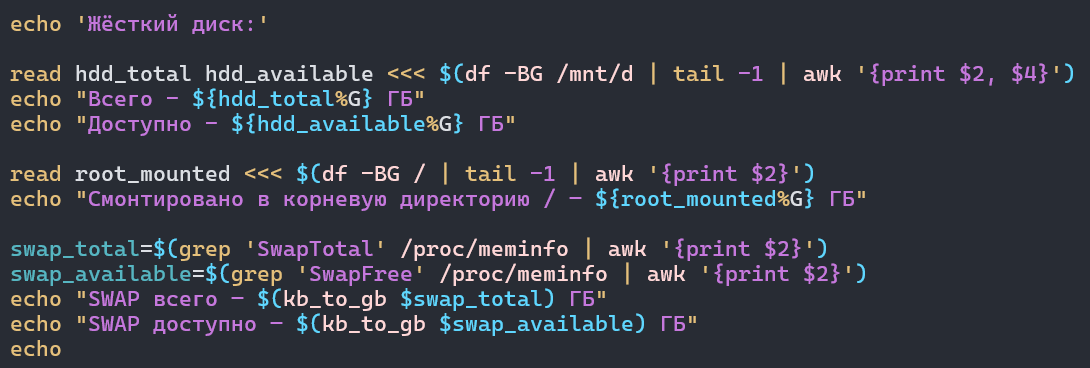
Рисунок 3. Оперативная память



Для вывода информации о жестком диске была использована команда read и оператор <<<, записывающий вывод команд справа в переменные слева. Необходимая информация, выводимая командой df с опциями -BG (вывод количества гигабайтов и округление в большую сторону) и путем /mnt/d (расположение жёсткого диска), была обрезана по последней строке командой tail, а awk выделил из нее поля (токены) 2 и 4 содержащие количество общего количество гигабайт и количество доступных гигабайт. Вывод количества гигабайт, смонтированных в корневую директорию /, был реализован аналогичных образом.

В случае с выводом общего и доступного размера swap (файла подкачки), был задействован ранее используемый файл /proc/meminfo и ранее созданная функция kb\_to\_gb для перевода килобайтов в гигабайты.

Рисунок 4. Жёсткий диск



Чтобы вывести количество сетевых интерфейсов, была использована команда wc с опцией -l, которая посчитала количество выводимых командой ls файлов (путей) в директории /sys/class/net. Тут следует отметить, что также была добавлена обработка исключений в виде перенаправления потока ошибок STDERR в «черную дыру» - 2>/dev/null, для того чтобы скрипт не прекратил работу в случае отсутствия сетевых интерфейсов.

Далее, для красивого табличного вывода информации обо всех интерфейсах, был создан общий блок кода, результат работы которого передавался команде column с опциями -t (табличный вывод) и -s ‘|’ (для разделения строчных токенов по разделителю ‘|’). В данном блоке кода, в цикле for происходил обход по всем сетевым интерфейсам, лежащим в папке /sys/class/net и составлялась единая строка со всей информацией, с соединением токенов посредством ‘|’. Для получения MAC-адресов, выводилось содержимое файла address, который лежит в директории каждого отдельного интерфейса. Для вывода IP-адресов, была задействована команда ip addr show, парсинг вывода которой после соответствующей обрезки выдавал искомое значение.

Так как два из 3-х имеющихся сетевых интерфейсов не являются реальными сетевыми адаптерами (docker0 и lo – localhost), то заместо в качестве показателя скорости им был установлен “N/A”. Для реализации подобной раздельной логики был задействован условный оператор if, и лишь при условии, что имя интерфейса равно ‘eth0’ (реальный сетевой адаптер), скорость рассчитывалась по-настоящему с помощью утилиты speedtest, заранее предустановленной на Linux.

Рисунок 5. Сетевые интерфейсы

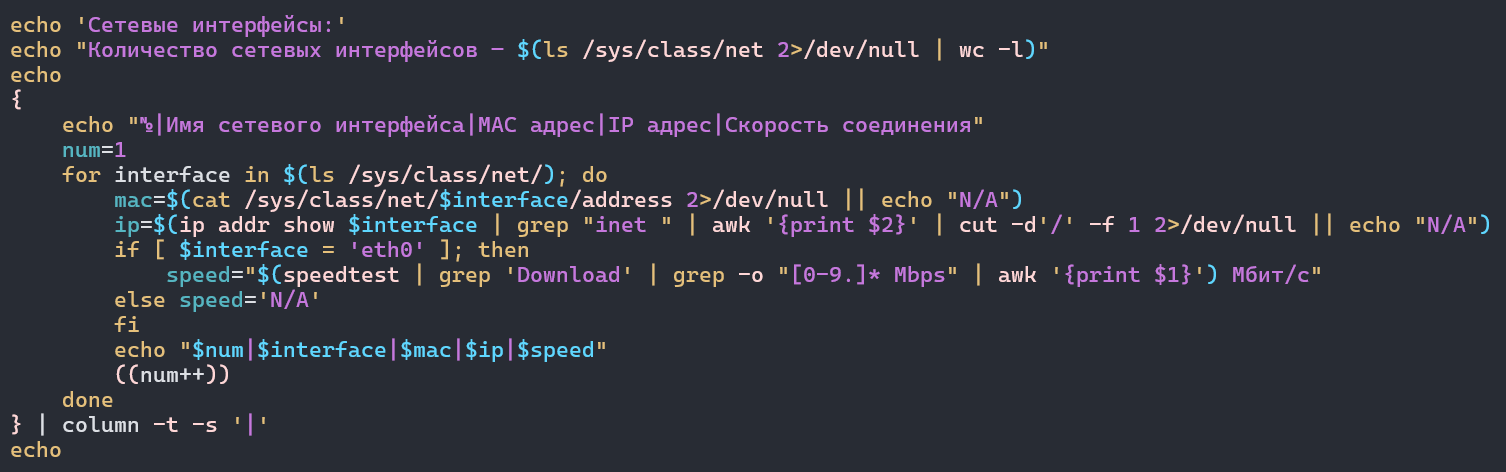
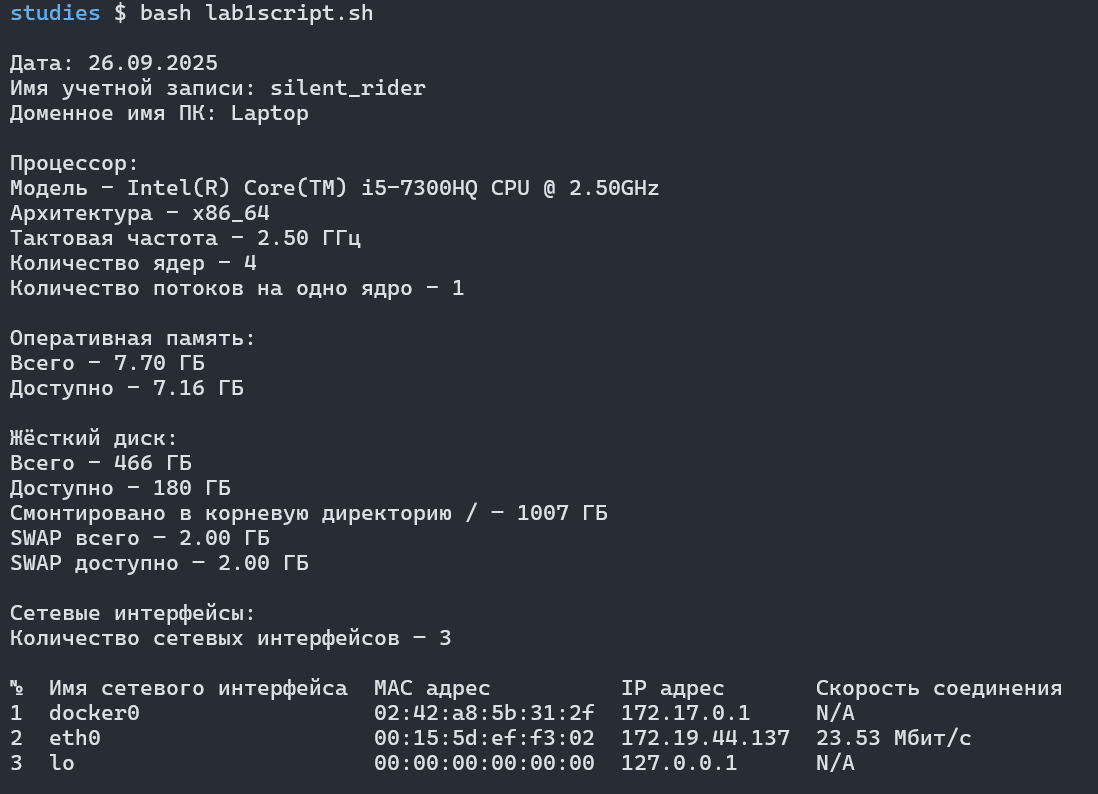


Рисунок 6. Результат работы скрипта



**Выводы**

В ходе лабораторной работы был разработан и протестирован bash-скрипт, автоматизирующий сбор и вывод ключевых характеристик персонального компьютера. Скрипт успешно извлекает информацию о дате выполнения, учётной записи пользователя, имени хоста, параметрах центрального процессора (модель, архитектура, тактовая частота, количество ядер и потоков), объёме оперативной памяти и swap, состоянии жёсткого диска, а также данных о сетевых интерфейсах, включая MAC- и IP-адреса. В результате проделанной работы были закреплены навыки работы с системными файлами Linux (/proc/cpuinfo, /proc/meminfo), использованы стандартные утилиты командной строки (grep, awk, cut, lscpu, df, ip), а также применены принципы структурированного вывода и обработки ошибок. Лабораторная работа позволила глубже понять архитектуру операционной системы Linux и механизмы получения аппаратных и сетевых характеристик программными средствами