##### МИНЦИФРЫ

##### «СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»

##### ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

##### «Разработать скрипт, который выводит информацию о системе»

##### по дисциплине «Архитектура вычислительных систем»

##### Выполнил: студент гр. ИП-212

##### Жеребцов Дмитрий Евгеньевич

##### Новосибирск 2024

# Постановка задачи

##### Задание.

Разработать скрипт, который выводит информацию о системе

* DoD на 3
  + Вывести название и версию ОС, версию и архитектуру ядра Linux
  + Вывести информацию о процессоре (Модель, частота, количество ядер, размер кэш-памяти)
  + Вывести информацию о размере оперативной памяти (Доступный размер, общий размер, использованный размер памяти)
* DoD на 4
  + DoD на 3
  + Вывести параметры (имя интерфейса, ip/mac) и скорость сетевого соединения
  + Вывести информацию о системных разделах (точка монтирования, размер раздела, занятое/свободное пространство)
* DoD на 5
  + DoD на 4
  + Обеспечить поддержку систем с более чем одним процессором (серверные системы, доступ к такой будет при необходимости) или более чем одним ip адресом на сетевом интерфейсе

# Выполнение работы

##### На первом этапе выполнения лабораторной работы — мне пришлось ознакомиться с интернет-ресурсами и *man* документацией команд, необходимых для получения информации о текущей системе пользователя.

##### Использовались как и команды bash так и файлы внутри операционной системы. (proc...)

#### Пример:

# Результат работы

##### Результат работы представлен на рисунке 1.

##### Рисунок 1. Итоговый вывод скрипта

# Исходный скрипт

get\_os\_info() {

os\_name=$(grep "NAME" /etc/os-release | head -1 | cut -d '=' -f 2)

os\_version=$(grep "VERSION" /etc/os-release | sed -n '2p' | awk -F'"' '{print $2}')

kernel\_version=$(uname -r)

kernel\_architecture=$(uname -m)

echo "Operating System: $os\_name"

echo "OS Version: $os\_version"

echo "Kernel Version: $kernel\_version"

echo "Kernel Architecture: $kernel\_architecture"

}

get\_cpu\_info() {

cpu\_model=$(grep "model name" /proc/cpuinfo | head -1 | cut -d ':' -f 2 | xargs)

cpu\_frequency=$(grep "cpu MHz" /proc/cpuinfo | head -1 | cut -d ':' -f 2 | xargs)

cpu\_cores=$(grep "cpu cores" /proc/cpuinfo | head -1 | cut -d ':' -f 2 | xargs)

cpu\_cache=$(lscpu -C | tail -n +2 | awk '{print "Name: ", $1, "Size: ", $2}')

cpu\_count=$(grep -c "physical id" /proc/cpuinfo)

echo "CPU Model: $cpu\_model"

echo "CPU Frequency: $cpu\_frequency MHz"

echo "Number of Cores: $cpu\_cores"

echo "Number of CPUs: $cpu\_count"

echo "Cache:"

echo "$cpu\_cache"

}

get\_memory\_info() {

total\_memory=$(free -m | grep Mem | awk '{print $2}')

available\_memory=$(free -m | grep Mem | awk '{print $7}')

used\_memory=$((total\_memory - available\_memory))

echo "Total Memory: $total\_memory MB"

echo "Available Memory: $available\_memory MB"

echo "Used Memory: $used\_memory MB"

}

get\_network\_info() {

interfaces=$(ip -o link show | awk -F': ' '{print $2}')

for iface in $interfaces; do

ip\_addrs=$(ip -o -4 addr show "$iface" | awk '{printf " %s", $4}')

mac\_addr=$(ip link show "$iface" | grep link/ether | awk '{print $2}')

speed=$(cat /sys/class/net/"$iface"/speed 2>/dev/null || nix run nixpkgs#iw "$iface" link | grep rx | awk '{print $3}')

echo "Interface: $iface"

echo "IP Addresses:$ip\_addrs"

echo "MAC Address: ${mac\_addr:-Unknown}"

echo "Speed: ${speed:-Unknown} Mbps"

echo

done

}

get\_partition\_info() {

echo "System Partitions:"

partitions=$(df -h --output=source,size,used,avail,target | grep "^/dev/")

echo """$partitions" | awk '{print "Mount Point:", $5, "Size:", $2, "Used:", $3, "Available:", $4}'

}

main() {

echo "System Information:"

get\_os\_info

echo ""

get\_cpu\_info

echo ""

get\_memory\_info

echo ""

get\_network\_info

get\_partition\_info

}

main