**Задание 1: Мониторинг использования памяти**

1. Просмотр информации о памяти

free -h

2. Интерактивное наблюдение за процессами и памятью

top

3. Сводка виртуальной памяти

vmstat

4. Установка и запуск htop

sudo apt-get install htop

htop

**Объяснение:**

* free -h показывает использование памяти в удобном для чтения формате.
* top обновляет данные в реальном времени.
* vmstat выводит сводку по CPU, памяти и вводу-выводу.
* htop удобен для управления процессами.

**Задание 2: Управление свопингом**

1. Проверка состояния свопа

swapon --show

2. Создание файла подкачки 1 ГБ

sudo fallocate -l 1G /swapfile

3. Настройка файла подкачки

sudo chmod 600 /swapfile

sudo mkswap /swapfile

4. Включение файла подкачки

sudo swapon /swapfile

5. Добавление в fstab для автоматического монтирования

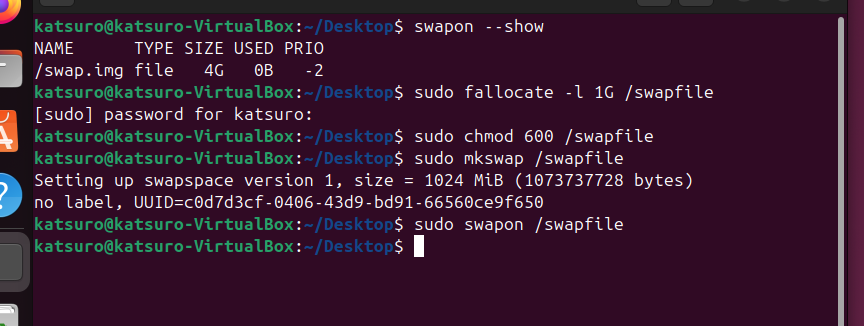
echo '/swapfile none swap sw 0 0' | sudo tee -a /etc/fstab

6. Проверка использования свопа

swapon --show

**Объяснение:**

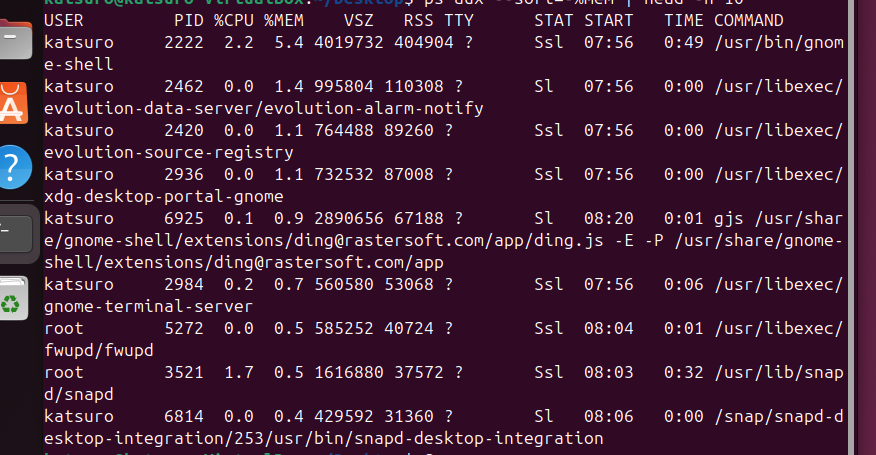
* fallocate создаёт файл заданного размера.
* mkswap делает файл подкачки.
* swapon активирует его.



**Задание 3: Анализ и оптимизация памяти**

1. 10 процессов с максимальным использованием памяти

ps aux --sort=-%mem | head -n 10

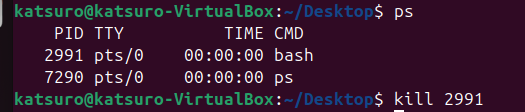


2. Подробная карта памяти процесса (замените <PID>)

pmap <PID>

3. Анализ данных

kill <PID>

clear

**Объяснение:**

* ps aux сортирует процессы по использованию памяти.
* pmap анализирует распределение памяти.

**Задание 4: Утилита для мониторинга использования памяти**

1. Создание нового файла скрипта

nano memory\_monitor.sh

2. Добавление кода в файл

#!/bin/bash

ps aux --sort=-%mem | head -n 10

3. Сохранение и установка прав на выполнение

chmod +x memory\_monitor.sh

4. Запуск скрипта

./memory\_monitor.sh

**Объяснение:**

* nano открывает текстовый редактор для создания файла.
* chmod +x делает файл исполняемым.
* Скрипт использует ps для анализа памяти.

**Задание 5: Настройка параметров ядра для управления памятью**

1. Открытие файла конфигурации sysctl

sudo nano /etc/sysctl.conf

2. Настройка параметров управления памятью

vm.swappiness=10

vm.dirty\_ratio=20

vm.dirty\_background\_ratio=10

3. Сохранение файла и применение изменений

sudo sysctl -p

**Объяснение:**

* vm.swappiness определяет частоту использования свопа.
* vm.dirty\_ratio задаёт процент памяти, занятой записываемыми страницами.
* vm.dirty\_background\_ratio запускает запись данных на диск при превышении заданного процента.

**Задание 6: Контроль за пределами использования памяти**

1. Установка предельного значения виртуальной памяти

ulimit -v 1024000

2. Запуск процесса, потребляющего память

# Например, запуск программы в пределах установленного лимита

./memory\_hungry\_program

3. Изменение предельного значения и повторный запуск

ulimit -v 2048000

./memory\_hungry\_program

**Объяснение:**

* ulimit -v задаёт лимит виртуальной памяти для процессов.

Задание 7: Анализ использования свопа

1. Установка smem

sudo apt-get install smem

2. Анализ использования свопа

smem

3. Идентификация процессов, использующих своп

smem -k

**Объяснение:**

* smem отображает использование памяти и свопа для процессов.

**Задание 8: Управление памятью с помощью cgroups**

1. Создание новой группы cgroups

sudo cgcreate -g memory:/test\_group

2. Установка предела использования памяти

echo 500M | sudo tee /sys/fs/cgroup/memory/test\_group/memory.limit\_in\_bytes

3. Запуск процесса в группе

sudo cgexec -g memory:/test\_group ./memory\_hungry\_program

4. Проверка использования памяти

cat /sys/fs/cgroup/memory/test\_group/memory.usage\_in\_bytes

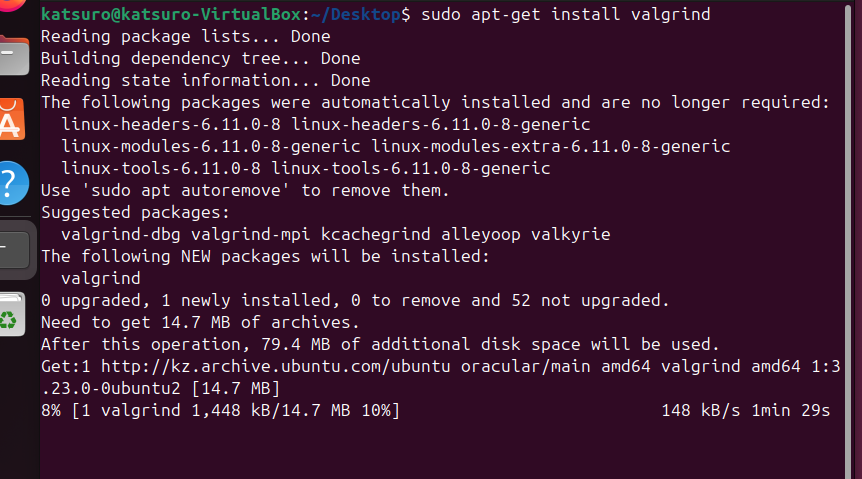
**Объяснение:**

* cgcreate создаёт группу cgroups.
* memory.limit\_in\_bytes задаёт лимит памяти.

**Задание 9: Диагностика утечек памяти с помощью Valgrind**

1. Установка Valgrind

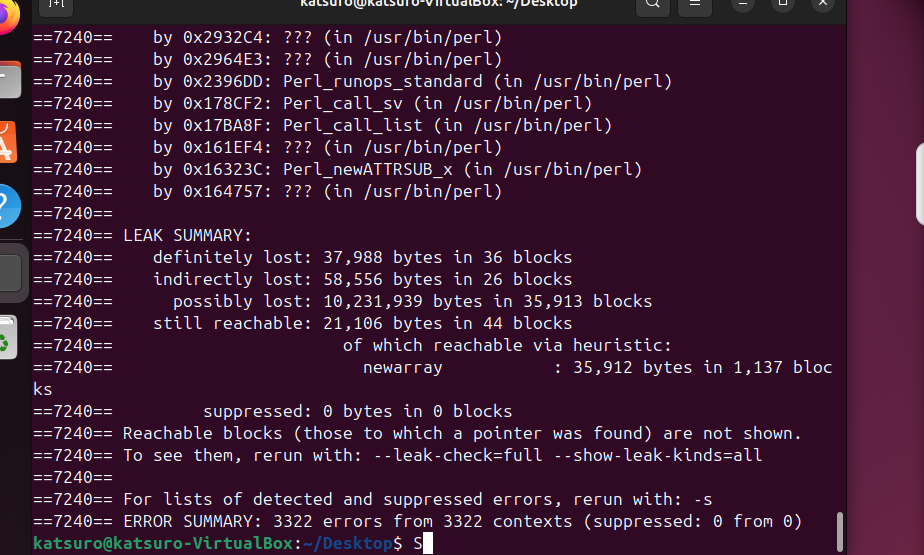
sudo apt-get install valgrind



2. Запуск приложения с Valgrind

valgrind --leak-check=full ./memory\_hungry\_program

3. Анализ результатов



**Объяснение:**

* valgrind анализирует приложение и обнаруживает утечки памяти.

**Задание 10: Мониторинг использования памяти с помощью watch и free**

1. Запуск команды watch

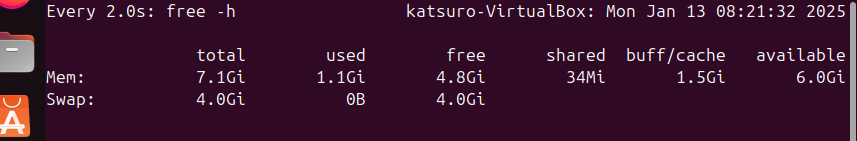
watch -n 2 free -h

2. Наблюдение за изменениями памяти

# Информация обновляется каждые 2 секунды.

**Объяснение:**

* watch периодически выполняет команду.
* free -h показывает информацию о памяти в реальном времени.



### **Вопросы для самопроверки:**

* **Как часто обновляется информация в команде watch?**Информация обновляется с интервалом, указанным с помощью флага -n. По умолчанию интервал составляет 2 секунды, если он не задан явно. Например, watch -n 2 обновляет данные каждые 2 секунды.
* **Какие изменения в использовании памяти вы наблюдаете?**При выполнении команды watch free -h можно наблюдать:
  + Увеличение или уменьшение доступной и используемой памяти.
  + Потребление памяти отдельными процессами, запуск новых приложений или завершение существующих.
  + Использование свопа при нехватке оперативной памяти.