Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа №1 по курсу**

**«Операционные системы»**

**Исследование системных вызовов и итоги других лабораторных работ**

Студент: Камеш Михаил Алексеевич

Группа: М8О–207Б–18

Вариант: 1

Преподаватель: Соколов Андрей Алексеевич

Оценка: \_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_

Москва, 2020.

**Содержание**

1. Постановка задачи
2. Краткой содержания лабораторных работ
3. Выводы

Цель работы

Приобретение практических навыков диагностики работы программного обеспечения.

Задание

При выполнении последующих лабораторных работ необходимо продемонстрировать ключевые системные вызовы, которые в них используются и то, что их использование соответствует варианту ЛР.

По итогам выполнения всех лабораторных работ отчет по данной должен содержать краткую сводку по исследованию последующих ЛР.

Список возможных стороннего программного обеспечения

1. Notepad

2. Notepad++

3. Powershell

4. Cmd

5. Paint

6. Shutdown

7. Fsutil

8. Cscript

9. И др.

Средства диагностики

Для ОС Windows

• Windbg

• http://windbg.info/doc/1-common-cmds.html

• SysinternalsSuite

• Handle.exe

• Procmon.exe

• Procexp.exe

Для ОС \*nix:

• strace

Краткое содержание лабораторных работ

lab 2.

Родительский процесс считывает две координаты передает их через канал дочернему процессу.

Дочерний процесс определяет к какой четверти относится точка, а далее передает результат родительскому процессу

Программа компилируется из одного файла main.c. В программе используются следующие системные вызовы:

1. read – для чтения данных из файла

2. write – для записи данных в файл

3. pipe – для создания однонаправленного канала, через который могут общаться два процесса. Системный вызов возвращает два дескриптора файлов. Один для чтения из канала, другой для записи в канал.

4. fork– для создания дочернего процесса.

5. close – для закрытия файла.

lab 3.

Реализовать битоническую сортировку, реализовать для неё многопоточность.

1. omp\_set\_dynamic - включает или отключает динамическую настройку количества потоков, доступных для выполнения последующих параллельных областей, путем установки значения ICV dyn-var.

2. omp\_get\_num\_procs - возвращает количество процессоров, доступных устройству.

3. omp\_set\_num\_threads - влияет на количество потоков, которые будут использоваться для последующих параллельных областей, которые не определяют предложение num\_threads, путем установки значения первого элемента nthreads-var ICV текущей задачи.

4. pthread\_create - функция используется для создания нового потока с атрибутами, указанными attr, в процессе

5. pthread\_join - функция должна приостанавливать выполнение вызывающего потока до завершения целевого потока, если целевой поток уже не завершился.

lab 4.

Родительский процесс считывает две координаты передает их через канал дочернему процессу.

Дочерний процесс определяет к какой четверти относится точка, а далее передает результат родительскому процессу

Использовать mmap.

1. mmap - создает новое отображение в виртуальном адресном пространстве вызывающего процесса.

2. close – для закрытия файла.

3. fork– для создания дочернего процесса.

4. sleep - переводит в неактивное состояние на определенный период времени.

lab 5.

Требуется создать динамическую библиотеку, которая реализует определенный функционал. Далее использовать данную библиотеку 2-мя способами:

1. Во время компиляции (на этапе «линковки»/linking)

2. Во время исполнения программы, подгрузив библиотеку в память с помощью системных вызовов

В конечном итоге, программа должна состоять из следующих частей:

• Динамическая библиотека, реализующая заданных вариантом интерфейс;

• Тестовая программа, которая используют библиотеку, используя знания полученные на этапе компиляции;

• Тестовая программа, которая использует библиотеку, используя только местоположение динамической библиотеки и ее интерфейс.

Провести анализ между обоими типами использования библиотеки.

1. dlclose - уменьшает количество открытых ссылок на библиотеку.

2. dlopen - открывает библиотеку и увеличивает количество открытых ссылок библиотеки.

3. dlsym - ищет в библиотеке символ (например, функцию) и возвращает его адрес.

4. dlerror - возвращает указатель на строку сообщения после возврата ошибки из dlopen (), dlsym () или dlclose ().

lab 6.

Реализовать распределенную систему по асинхронной обработке запросов.

В данной распределенной системе должно существовать 2 вида узлов: «управляющий» и «вычислительный».

Необходимо объединить данные узлы в соответствии с той топологией, которая определена вариантом.

Связь между узлами необходимо осуществить при помощи технологии очередей сообщений.

Также в данной системе необходимо предусмотреть проверку доступности узлов в соответствии с вариантом. (2 - 4 - 1)

0mq – библиотека для С++, в которой возможна реализация передачи данных по TCP протоколу.

Выводы:

Был выполнен ряд работ на языке С по курсу Операционные системы. Для написания использовалась Ubuntu (unix) и разные систимные вызовы, работающие в данной системе.

Были получены навыки работы с процессами и потоками операционных систем и способы их манипулирования. Для анализа работ задач использовалась команда strace.