

Лабораторные ОС 2023

Лабораторная работа №1 «Характеристики Операционных систем»

Разработать программу, которая мониторит состояние загрузки центрального процессора, оперативной памяти и файловой системы. Программа должна строить графики.

Так же программа должна показывать информацию о компьютере:

- Архитектура процессора, количество процессоров, тип процессора
- имя компьютера, имя пользователя
- информация об устройствах

Для реализации использовать следующие функции:

WinAPI	POSIX
GetSystemTimes	парсинг директорий
GlobalMemoryStatys	/proc/cpuinfo
GlobalMemoryStatysEx	/proc/stat
GetSystemInfo	/proc/uptime
GetDiskFreeSpace	/proc/loadavg

Лабораторная работа №2 «Работа с процессами и потоками в Операционной Системе»

Разработать диспетчер процессов и потоков, который реализует следующие функции:

- Создание нового процесса, с заданием начального приоритета;
- Завершение существующего процесса и дерева процессов;
- Приостановить-возобновить поток выбранного процесса;
- Отображение всех загруженных процессов с их ID в виде дерева;
- Информация о процессе (количество занимаемой памяти, используемые модули, количество потоков);
- отображение и изменение приоритетов у выбранного процесса;

Для реализации использовать следующие функции:

WinAPI	POSIX
CreateProcess	fork exec*
OpenProcess	
TerminateProcess	kill
EnumProcesses	sysctl(KERN_PROC_ALL)

EnumProcessModules	
GetModuleBaseName	getpwuid
GetModuleInformation	
WaitForSingleObject	Waitpid
Thread32First, Thread32Next	
SetPriorityClass	pthread_attr_setschedpolicy setpriority nice
GetPriorityClass	pthread_attr_getschedpolicy getpriority
SetThreadPriority	pthread_attr_setschedparam
GetThreadPriority	pthread_attr_getschedparam
ResumeThread	
SuspendThread	
GetThreadId	
OpenThread	
TerminateThread	
CreateToolhelp32Snapshot	
Process32First, Process32Next	
Module32First, Module32Next	
GetProcessMemoryInfo	

Лабораторная работа №3 **«синхронизация процессов и потоков»**

Реализовать одну из задач с использованием критических секций, семафоров или мьютексов (WinAPI) или семафоров или мьютексов (POSIX)

Задача обедающих философов:

Пять философов сидят за круглым столом, и у каждого из них есть тарелка спагетти. Эти спагетти настолько скользкие, что есть их можно только двумя вилками. Между каждыми двумя тарелками лежит одна вилка. Жизнь философа состоит из чередующихся периодов приема пищи и размышлений. Когда философ становится голоден, он старается поочередно в любом порядке завладеть правой и левой вилками. Если ему удастся взять две вилки, он на некоторое время приступает к еде, затем кладет обе вилки на стол и продолжает размышления. Основной вопрос состоит в следующем: можно ли написать программу для каждого философа, который действует предполагаемым образом и никогда при этом не попадает в состояние зависания?

Задача спящего брадобрея:

В парикмахерской есть один брадобрей, его кресло и n стульев для посетителей. Если желающих воспользоваться его услугами нет, брадобрей сидит в своем кресле и спит. Если в парикмахерскую приходит клиент, он

должен разбудить брадобрея. Если клиент приходит и видит, что брадобрей занят, он либо садится на стул (если есть место), либо уходит (если места нет). Необходимо запрограммировать брадобрея и посетителей так, чтобы избежать состояния состязания.

Задача о курильщиках:

Изначально есть три заядлых курильщика, сидящих за столом. Каждому из них доступно бесконечное количество одного из трёх компонентов: у одного курильщика — табака, у второго — бумаги, у третьего — спичек. Для того чтобы делать и курить сигары, необходимы все три компонента. Также, кроме курильщиков, есть бармен, помогающий им делать сигареты: он недетерминированно выбирает двух курильщиков, берёт у них по одному компоненту из их запасов и кладёт их на стол. Третий курильщик забирает ингредиенты со стола и использует их для изготовления сигареты, которую он курит некоторое время. В это время бармен, увидев стол пустым, снова выбирает двух курильщиков случайным образом и кладёт их компоненты на стол. Процесс повторяется бесконечно. Курильщики, по условию проблемы, честные: они не прячут компоненты, выданные барменом, — они лишь скручивают сигарету тогда, когда докурят предыдущую. Если бармен кладёт, например, табак и бумагу на стол, пока поставщик спичек курит, то табак и бумага останутся нетронутыми на столе, пока курильщик со спичками не докурит сигарету и только затем не возьмёт табак и бумагу.

Для реализации использовать следующие функции :

WinAPI

POSIX

InitializeCriticalSection

WaitForSingleObject

DeleteCriticalSection

LeaveCriticalSection

EnterCriticalSection

_beginthreadex

CreateMutex

ReleaseMutex

OpenMutex

CreateSemaphore

ReleaseSemaphore

OpenSemaphore

WaitForMultipleObjects

pthread_mutex_init

pthread_mutex_unlock

Semget

Semop

Semget

pthread_mutex_lock

semop

Лабораторная работа №4 «межпроцессное взаимодействие»

Реализовать простую клиент серверную программу с использованием именных каналов - Логирование (сбор данных от клиентов о работе программы).

Сделать как минимум двух клиентов (текстовый процессор и вычислительный процессор).

Реализовать клиента который посылает сообщения в равные промежутки, и по изменению действия.

Сообщения типа: начало работы, конец работы, набор текста или вычисления и простой.

Сервер получает информацию и сохраняет её в Базе данных.

БД должна иметь как минимум 2 таблицы: таблица о клиентах и таблица о событиях.

Так же на сервере должна быть возможность запрашивать среднии характеристики заданного клиента (входные параметры - имя клиента, временной промежуток и параметр расчета. Например узнать сколько работал блокнот с 01.01.2023 по 03.01.2023 в среднем за минуту или за час).

Для реализации использовать следующие функции :

WinAPI

POSIX

CreateFile

WriteFile

ReadFile

CloseHandle

close

msgetr

CreateNamedPipe

mkfifo

msgget

CreateThread

pthread_create

ConnectNamedPipe

DisconnectNamedPipe

WaitNamedPipe

TransactNamedPipe

CallNamedPipe

PeakNamedPipe

Лабораторная работа №5 «файловая система»

Реализовать виртуальную файловую систему со следующими основными функциями:

- CD [path] сделать path текущим каталогом

- DIR вывести содержимое текущего каталога на экран
- MOUNT [source] [target] создать копию реальной файловой системы, задаваемой путем source, в виртуальной системе target
- UMount [path] удалить каталог
- QUIT выход

Виртуальная файловая система не должна изменять существующую файловую систему.

Также реализовать возможность сохранения и считывания виртуальной файловой системы из файла.

Варианты заданий:

- 1 Вариант 1 (работа с файлами)
 - RENAME file1 file2 переименовать file1 в file2
 - DEL file удаление файла
 - COPY file path копирование файла
- 2 Вариант 2 (работа с каталогом)
 - DIR mask вывод по маске
 - SIZEOF path вывод размера каталога
 - FIND mask найти файлы по маске
- 3 Вариант 3 (работа с файлом)
 - INFO file вывод информации о файле
 - PRINT file вывод файла на экран
 - FINDSTR string file поиск строки в файле