Лабораторная работа №2. Основы процессов в UNIX.

Методические указания (aka ссылочки):

- https://www.freebsd.org/doc/ru_RU.KOI8-R/books/faq/
- https://www.freebsd.org/doc/ru_RU.KOI8-R/books/handbook/
- http://ru.manpages.org/signal/7
- http://manpages.org/pthreads/7
- https://www.geeksforgeeks.org/posix-shared-memory-api/
- https://habr.com/ru/post/326138/

Работу разрешается выполнять на языках программирования C/C++, .NET Core Platform, JVM Platform, другой по согласованию с преподавателем.

Задание на лабораторную работу:

1. Разверните на локальной машине образ ОС FreeBSD 12.0;
2. При необходимости установите необходимые инструменты разработки и среды исполнения;
3. Изучите документацию к стандарту POSIX;
4. Спроектируйте и реализуйте систему мониторинга и управления удаленным сервером, отвечающую следующим требованиям:
    4.1. система построена в соответствии с клиент-серверной архитектурой;
    4.2. коммуникация между клиентской и серверной частями организуется через сокеты с помощью протоколов UDP/TCP/HTTP/или на ваш обоснованный выбор;
    4.3. серверная часть системы разбита на отдельные процессы, реализующие строго ограниченный набор возможностей;
    4.4. коммуникация между компонентами (процессами) серверной части системы организована с помощью каналов (pipe) и сигналов (signal);
    4.5. серверная часть системы реализует следующие возможности:
        4.5.1. запуск процессов с заданным именем, параметрами и от имени/UID заданного пользователя, если система имеет на это привилегии:

4.5.1.1. запуск процесса в фоне, при этом вывод процесса не отправляется на клиентскую часть, но добавляется в отдельный лог-файл;

4.5.1.2. запуск и ожидание завершения работы процесса, при этом пользователю отправляется вывод процесса и код возврата;

4.5.2. получение текущей информации о состоянии процессов, аналогично работе программы ps;

4.5.3. мониторинг состояния процессов с заданным именем/PID:

4.5.3.1. за каждым заданным для наблюдения процессом следит отдельный процесс системы мониторинга;

4.5.3.2. при завершении наблюдаемого процесса, наблюдающий процесс регистрирует изменение состояния процесса, и также завершается с записью сообщения о своем завершении в лог-файл;

4.5.3.3. за каждым параметром наблюдаемого процесса, по возможности, следит отдельный поток наблюдающего процесса;

4.5.3.4. при работе нескольких потоков должна сохраняться когерентность данных о состоянии наблюдаемого процесса;

4.5.3.5. история изменения состояний процессов должна быть доступна для получения с клиентской части не менее чем в течение 5 минут;

4.5.3.6. сообщения об изменении состояния процесса сохраняются в лог-файл;

4.5.3.7. (*) система реализует интерфейс длинных запросов (Long poll), благодаря которому клиентская часть может получить уведомление об изменении состояния процесса мгновенно;

4.5.4. система следит за запуском/завершением всех процессов, несмотря на то, что они могут быть не выбраны по имени/PID; при запуске/остановке не наблюдаемых процессов в лог-файл добавляется сообщение о запуске/остановке процесса и его имени/PID;

4.5.5. рассылка процессам с выбранным именем/PID/PPID сигналов с указанным именем/номером;

4.5.6. для ведения лог-файлов должен быть выделен отдельный процесс, получающий сообщения от других компонентов (процессов) системы мониторинга с помощью разделяемых регионов памяти (shared memory); прямая работа с лог-файлом из других компонентов системы запрещена;

4.5.7. при получении корневым процессом системы сигнала SIGTERM, система должна корректно завершать работу, при этом каждый дочерний процесс, помимо освобождения ресурсов должен сообщить в лог-файл о своем завершении;

4.5.8. получение текущего состояния машины, на которой запущена система мониторинга (размер страницы виртуальной памяти/объем свободной и занятой памяти/количество физической памяти/модель процессора/*);

4.5.9. простая аутентификация по логину и паролю;

4.6. клиентская часть реализует следующие возможности:

4.6.1. получение команд от пользователя и отправка запросов к серверной части системы мониторинга;

4.6.2. отображение информации, полученной от серверной части в человекочитаемом виде;

4.6.3. (*) режим подписки на уведомления от серверной части системы мониторинга (Long poll);

4.6.4. клиентская часть может быть выполнена как набор отдельных программ, либо в виде одного интерактивного приложения (можно оконного, можно ncurses);

4.7. ограничения на разрабатываемую систему:

4.7.1. не допускается использование функций и классов библиотек для ввода-вывода (в том числе, файлового), создания потоков, семафоров, мьютексов, и т.д. если явно не указано обратное (запрещены file, fstream, mutex, semaphore, thread, process, system и т.д.);

4.7.2. допускается использование GTK, Qt, ncurses и подобных библиотек исключительно для создания пользовательского интерфейса;

4.7.3. допускается разработать свои классы и функции-обертки для системных функций, реализующих необходимые возможности по взаимодействию с системой;

4.7.4. допускается использование библиотек, направленных на обработку данных и не взаимодействующих с системой (разбор или создание HTTP запросов при необходимости, сам запрос отправляется вашими силами, используя системные функции или ваши обертки);

4.7.5. запрещается использовать сторонние программы для получения информации о системе и управления ей;

4.7.6. запрещается использовать веб-фреймворки (Spring, ASP.NET, Boost, Casablanca и подобные);

5. Поэтапно публикуйте результаты вашей работы на GitHub;
6. Репозиторий должен содержать файлы документации (желательно MarkDown), в которых содержится информация об API системы, ее архитектуре (ОЧЕНЬ желательны диаграммы), командах а также руководство пользователя;

Рекомендации:

Будут пополняться