Задачи к курсу “Операционные системы” ФИТ НГУ, 1-й семестр.Draft

Правила сдачи задач

При подготовке задачи к сдаче убедитесь, что:

1. Код оформлен в едином стиле:
    a. однообразное именование функций, типв, переменных, макросов и т.д. (при этом допускается, что для каждого из этих объектов стиль именования свой);
    b. однообразные отступы;
    c. однообразная расстановка скобок;
    d. и т.д..
2. Проверены возвращаемые значения функций и системных вызовов и предусмотрена адекватная реакция на ошибки. Можно не проверять на возвращаемое значение функцию printf().
3. Все ресурсы выделенные явно, должны быть также явно освобождены. Не должно быть точек выхода из программы, где не освобождаются явно выделенные ресурсы.
4. Компиляция должна проходить без ошибок и предупреждений на максимальном уровне предупреждений.
5. Программа делает то что требуется.

Обязательно подготовьтесь к ответам на вопросы по той теме, на которую рассчитана задача.

В процессе сдачи, преподаватель может также потребовать изменить, добавить какую-либо функциональность или провести программный эксперимент.

Критерии оценки

На оценку “удовлетворительно” надо сдать по одной задаче из каждого раздела.
На оценку “хорошо” - по две задачи из каждого раздела.
На оценку “отлично” - по три задачи из каждого раздела.

Задачи

Компиляция, сборка, запуск

1. Написать программу hello.c, которая выводит фразу “Hello world”:
    a. получить исполняемый файл;
    b. посмотреть unresolved symbols (puts, printf) с помощью nm;

    c. посмотреть зависимости (ldd);
    d. запустить.
2. Написать статическую библиотеку с функцией hello_from_static_lib() и использовать ее в hello.c:
    a. посмотреть исполняемый файл на предмет того будет ли функция hello_from_static_lib() unresolved. Почему?
    b. где находится код этой функции?
3. Написать динамическую библиотеку с функцией hello_from_dynamic_lib() и использовать ее с hello.c:
    a. посмотреть состояние функции hello_from_dynamic_lib в получившимся исполняемом файле. Объяснить увиденное.
4. Написать динамическую библиотеку с функцией hello_from_dyn_runtime_lib() и загрузить ее в hello.c с помощью dlopen(3). Объяснить что происходит.

Системные вызовы

1. Проведите следующие эксперименты:
    a. запустите программу hello world из предыдущей задачи под strace:
        i. обратите внимание какие системные вызовы были вызваны в процессе исполнения программы. Чем обусловлено такое количество системных вызовов. Какой системный вызов используется для вывода “hello world”? Изучите этот вызов и разберитесь что он принимает и возвращает.
        ii. используйте этот сискол в программе hello world вместо printf(). Убедитесь что этот вызов присутствует в выводе strace.
        iii. напишите свою обертку над этим сисколом. Для этого используйте функцию syscall() из libc. Также проверьте вывод strace.
    b. Запустите под strace команду ‘wget kernel.org’ (если нет wget, используйте curl). Получите статистику использования системных вызовов порожденным процессом.
2. Разберитесь как устроена функция syscall(). Напишите код, который напечатает hello world без использования функции syscall().
3. Разберитесь как работает системный вызов ptrace(2) и напишите программу, которая породит процесс и выведет все системные вызовы дочернего процесса. (Можно решить эту задачу после изучения темы “Процессы”).

Файлы и Файловые системы

1. Написать программу, которая копирует каталог “задом наперед”. Программа получает в качестве аргумента путь к каталогу. Далее:
    a. Программа создает каталог с именем заданного каталога, прочитанного наоборот. Если задан каталог “qwerty”, то должен быть создан каталог “ytrewq”.
    b. Программа копирует все регулярные файлы из исходного каталога в целевой (пропуская файлы другого типа), переворачивая их имена и

содержимое. То есть с именами файлов поступаем также как и с именем каталога, а содержимое копируется начиная с последнего байта и до нулевого.

2. Написать программу, которая создает, читает, изменяет права доступа и удаляет следующие объекты: файлы, каталоги, символьные и жесткие ссылки. Для определения того какая именно функция должна быть исполнена предлагается иметь необходимое количество жестких ссылок на исполняемый файл с именами соответствующими выполняемому действию и в программе выполнять функцию соответствующую имени жесткой ссылки. Программа должна уметь:
    a. создать каталог, указанный в аргументе;
    b. вывести содержимое каталога, указанного в аргументе;
    c. удалить каталог, указанный в аргументе;
    d. создать файл, указанный в аргументе;
    e. вывести содержимое файла, указанного в аргументе;
    f. удалить файл, указанный в аргументе;
    g. создать символьную ссылку на файл, указанный в аргументе;
    h. вывести содержимое символьной ссылки, указанный в аргументе;
    i. вывести содержимое файла, на который указывает символьная ссылка, указанная в аргументе;
    j. удалить символьную ссылку на файл, указанный в аргументе;
    k. создать жесткую ссылку на файл, указанный в аргументе;
    l. удалить жесткую ссылку на файл, указанный в аргументе;
    m. вывести права доступа к файлу, указанному в аргументе и количество жестких ссылок на него;
    n. изменить права доступа к файлу, указанному в аргументе.

3. Написать программу, которая выводит содержимое /proc/pid/pagemap

Процессы

В разработке

Понятие пользователя. Управление правами.

В разработке

Межпроцессное взаимодействие

В разработке

Сеть

В разработке