Содержание

1 Методика оценивания

Зачёт ставится при условии выполнения 10 практических работ. Оценка зависит от сроков сдачи и количества сданных работ. При сдаче 10 работ оценка не выше «удовлетворительно»; 12 работ – не выше «хорошо». Оценка за каждую работу зависит от срока сдачи работы Работа не должна содержать ошибок (в том числе должны проверяться все возможные ошибки выполнения вызываемых функций).

Все работы, кроме 8 практической работы, выполняются на языке `C` или `C++` в среде той операционной системы, что указана в работе. Допустимо использование вместо полноценного дистрибутива `Linux` эмулятора в случае, если он поддерживает возможность использования всех функций, перечисленных в описаниях практических работ.

2 Практическая работа №1. Работа с файловой системой в Linux (12 часов)

Срок сдачи: 28 сентября

Цель работы: повторение языка C/C++; повторение среды Linux; ознакомление с фукциями, предназначенными для работы с каталогами.

При выполнении практических работ действуют те же правила, что и в прошлом году (касательно качества кода).

Все работы делаются на языке C/C++ (кроме специально оговоренных случаев). Трансляторы: gcc/clang (на ваш выбор или выбор администраторов колледжа).

Работы выполняются в среде ОС Linux

На первых порах все задания 0-го уровня.

В заданиях запрещено вызывать команды shell (bash) или использовать другие способы их выполнения, кроме использования функций Posix или близких к ним (exec кроме явно указанных случаев, в частности, запрещен).

Время на первую работу – 3 пары: 1 пара на воспоминания языка Си, одна пара на изучение интерфейса Posix и смежных с ним, одна пара на реализацию.

Функции, которые необходимо изучить: rename (stdio.h), opendir, closedir, readdir, scandir (dirent.h), unlink (unistd.h), stat (sys/stat.h). При приеме работы будет проверено знание этих функций (наизусть запоминать поля и параметры не нужно).

Все ошибки необходимо обрабатывать, способ обработки оставляется на усмотрение студента: в некоторых случаях надо вывести диагностическое сообщение и завершить работу программы; в некоторых случаях – вывести сообщение, но продолжить выполнение.

Номера работ выбираются в соответствии с номером в журнале третьего курса.

Все файлы, с которыми происходит работа, лежат в текущем каталоге.

1. Удалить первый символ у имен обычных файлов (regular file) текущего каталога в случае, если в имени ровно 2 гласных латинских буквы.

2. Удалить обычные файлы (regular file), имена которых читаются слева направо и справа налево одинаково
3. Перед каждым именем файла (regular file) вставить его номер (1, 2, 3 и т. д.). Номер выставляется в соответствии с тем, в каком порядке выдает результаты функция readdir.
4. У всех имен файлоа (regular file) заменить первый символ на заглавную латинскую буквы в случае, если этот символ – строчная буква и других символов, кроме латинских букв в имени файла нет.
5. Удалить обычные файлы (regular file), имена которых имеют в своем составе ровно 2 согласных латинских буквы.
6. Во всех обычных файлах (regular file), имена которых состоят только из латинских букв, поменять порядок букв на противоположный.
7. Удалить обычные файлы (regular file), у которых расширение и основная часть имени совпадают.
8. У обычных файлов (regular file), у которых имя является целым неотрицательным числом прибавить к этому числу 1000. Предполагается, что изначально файлы имеют названия, числовое значение которых не превышает 999.
9. Удалить из имен обычных файлов (regular file) все цифры.
10. Удалить обычные файлы (regular file), длина которых совпадает с названием файла (в случае, если название файла – это целое неотрицательное число).
11. Переименовать обычные файлы (regular file), название которых содержит символ `!` в имя, совпадающее с его длиной. Противоречия этого задания решаются так, как указано выше *курсивом*.
12. К имени обычных файлов (regular file), содержащим символ подчеркивания, приписать после этого подчеркивания длину файла. Противоречия этого задания решаются так, как указано выше *курсивом*.
13. К имени обычных файлов (regular file), имеющих нечетную длину, приписать символ `!`, если он отсутствует.
14. Удалить обычные файлы (regular file), имя которых содержит символ `~` в случае, если длина их на 0.
15. н Удалить обычные файлы (regular file), имя которых является целым неотрицательным числом и превышает размер файла.
16. Удалить у всех имен обычных файлов (regular file) знаки `~`, размещенные в конце имени.
17. Удалить все обычные файлы (regular file), у которых права доступа владельца включают возможность их запуска.
18. Заменить пробелы в именах всех обычных файлов (regular file) на знаки подчеркивания.
19. Удалить из имен обычных файлов (regular file) гласные буквы латинского алфавита.
20. Переименовать обычные файлы (regular file) таким образом, чтобы чередовались заглавные и строчные буквы латинского алфавита в именах файлов (в именах файлов меняются только латинские буквы, остальные символы не меняются). После любого символа, не являющегося латинской буквой, следующая за ним латинская буква – заглавная.
21. Переставить символы в основной части всех имен обычных файлов (не в расширении) в обратном порядке.
22. Все обычные файлы, имеющие нулевую длину переименовать в имя, имеющее формат: `zero<номер>`, где `<номер>` – это числа попорядку, начиная с 1.
23. Удалить обычные файлы, которые в имени имеют хотя бы два символа, не являющиеся латинской буквой или цифрой.
24. В именах обычных файлов заменить символ `$` на длину файла.
25. Переименовать обычные файлы, к которым есть доступ только на чтение для владельца путем добавления к нему расширения `.read`
26. Переименовать обычные файлы, имена которых – целые неотрицательные числа на название того же числа в 16-ой системе счисления.

3 Практическая работа №2. Работа с файловой системой в Windows (4 часа)

Срок сдачи: 5 октября

Цель работы: ознакомление с фукциями, предназначенными для работы с каталогами (WinAPI).

Функции: FindFirstFile, FindNextFile, DeleteFile, MoveFile, DeleteFile (windows.h)

Выполните задание практической работы №1 с использованием Visual Studio/C++/консольное приложение и функций WinAPI.

4 Практическая работа №3. Рекурсия, файловая система в Linux (4 часа)

Срок сдачи: 12 октября

Цель работы: реализация рекурсивных алгоритмов

Выполните задание практической работы №1 с учетом того, что файлы ищутся не только в текущем каталоге, но и во всех его подкаталогах (любой степени вложенности). Обратите внимание на то, что существуют фиктивные каталоги `.` и `..` в каждом каталоге.

5 Практическая работа №4. Рекурсия, файловая система в Windows (2 часа)

Срок сдачи: 19 октября

Цель работы: реализация рекурсивных алгоритмов

Выполните задание практической работы №3 с учетом того, что файлы ищутся не только в текущем каталоге, но и во всех его подкаталогах (любой степени вложенности). Обратите внимание на то, что существуют фиктивные каталоги `.` и `..` в каждом каталоге.

6 Практическая работа №5. Работа с параметрами командной строки (2 часа)

Срок сдачи: 19 октября

Цель работы: изучение способа обработки аргументов из командной строки

Напишите программу, работающую в среде Linux и осуществляющую обработку параметров командной строки согласно своему варианту. При отсутствии аргументов программа должна выдавать инструкцию, при неверном числе аргументов – информацию об ошибке.

Темы для изучения: аргументы argc, argv функции main

1. Напишите программу mycp, имеющую два аргумента и копирующая файл, имя которого стоит в первом аргумент, в файл, имя которого стоит во втором аргументе. Пример вызова: ./mycp test1 test2
2. Напишите программу myrm, имеющую один аргумент и удаляющую файл, имя которого задано. Пример вызова: ./myrm test.txt
3. Напишите программу mycat, имеющую один аргумент и выводящую содержимое файла на экран. Пример вызова: ./mycat test.txt
4. Напишите программу mytail, имеющую два аргумента: имя файла и число n – и выводящую содержимое последних n строк файла. Пример вызова: ./mytail test.txt 10
5. Напишите программу myhead, имеющую два аргумента: имя файла и число n – и выводящую содержимое первых n строк файла. Пример вызова: ./myhead test.txt 10
6. Напишите программу mymkdir, имеющую один аргумент и создающую каталог. Пример вызова: ./mymkdir dir1
7. Напишите программу mynl, имеющую один аргумент и выводящую содержимое файла с указанием номеров строк. Пример вызова: ./mynl test.txt
8. Напишите программу mywc, имеющую один аргумент и выводящую количество строк, слов и символов в файле.
9. Напишите программу mymv, имеющую два аргумента и перемещающая файл, имя которого стоит в первом аргумент, в файл, имя которого стоит во втором аргументе. Пример вызова: ./mymv test1 test2
10. Напишите программу exists, имеющую один аргумент – имя файла, и выводящую на экран 1, если файл существует и 0 – если нет. Пример вызова: ./exists test.txt
11. Напишите программу np2n, имеющую два аргумента – имена файлов, которая в первом файле последовательность байтов 13 и 10 заменяет на байт 10 и результат пишет во второй файл. Пример вызова: ./np2n test.txt test2.txt
12. Напишите программу n2np, имеющую два аргумента – имена файлов, которая в первом файле байт 10 заменяет на байт 13 и 10 и результат пишет во второй файл. Пример вызова: ./n2np test.txt test2.txt
13. Напишите программу dec2hex, имеющую один аргумент – десятичное натуральное число и выводящее соответствующее 16-ое число. Пример вызова: ./dec2hex 125
14. Напишите программу hex2dec, имеющую один аргумент – 16-ое натуральное число и выводящее соответствующее 10-ое число. Пример вызова: ./hex2dec F2F
15. Напишите программу isDir, имеющую один аргумент – имя файла и выводящая 1, если файл - каталог и 0 в противном случае. Пример вызова: ./isDir directory

16. Напишите программу zero, имеющую один аргумент – имя файла и заменяющую указанный файл на файл нулевой длины. Пример вызова: ./zero test.dat

17. Напишите программу exchange, имеющую два аргумента – имена файлов и меняющаю эти файлы местами. Пример вызова: ./exchange text1.txt file1.txt

18. Напишите программу cat2, имеющую три аргумента – имена файлов, которая содержимое первых двух файлов записывает подряд в третий файл. Пример вызова: ./cat2 file1.dat file2.dat result.dat

19. Напишите программу rev, имеющую один аргумент – имя файла, которая выводит содержимое файла, в котором каждую строку переворачивает наоборот (располагает символы в обратном порядке). Пример вызова: ./rev test.txt

20. Напишите программу sort, имеющую один аргумент – имя файла, которая выводит содержимое файла в алфавитном порядке (сортирует строки). Пример вызова: ./sort test.txt

21. Напишите программу uniq, имеющую один аргумент – имя файла, которая выводит все строки файла, исключая повторяющиеся строки (в любом порядке). Пример вызова: ./uniq test.txt

22. Напишите программу mysubst, имеющую три аргумента: строку s1, строку s2 и имя файла, которая выводит содержимое файла, заменяя подстроки s1 на s2. Пример вызова: ./mysubst Миша Мишутка test.txt

23. Напишите программу mytruncate, имеющую три аргумента: число l и название файла, которая данный файл обрезает, уменьшая его размер до l. Пример вызова: ./mytruncate 10 test.txt

24. Напишите программу mydd, имеющую три аргумента: число l и название двух файлов, которая копирует из первого файла во второй l байт. Пример вызова: ./mydd 10 test.txt test2.txt

25. Напишите программу myunexpand, имеющую два аргумента – имена двух файлов, которая в данном первом файле заменяет несколько пробелов, оканчивающихся на позициях, пропорциональных 8, на символ табуляции и выводит результат во второй файл. Пример вызова: ./myunexpand test.txt test2.txt

26. Напишите программу myexpand, имеющую два аргумента – имена двух файлов, которая в данном первом файле заменяет табуляции на пробелы, оканчивающиеся на позиции, пропорциональной 8, и выводит результат во второй файл. Пример вызова: ./myexpand test.txt test2.txt

7 Практическая работа №6. Создание новых процессов в Linux (4 часа)

Срок сдачи: 26 октября

Цель работы: изучение способа создания новых процессов.

Реализуйте задание 3-ей практической работы таким образом, чтобы каждый каталог обрабатывался отдельным процессом.

Темы для изучения: функция fork

8 Практическая работа №7. Создание новых потоков в Windows (10 часов)

Срок сдачи: 16 ноября

Цель работы: изучение способа создания новых потоков.

Реализуйте задание 4-ей практической работы таким образом, чтобы каждый каталог обрабатывался отдельным процессом.

Темы для изучения: функция CreateThread, WaitForSingleObject, CloseHandle

9 Практическая работа №8. Создание новых нитей выполнения в LCL/VCL (6 часов)

Срок сдачи: 23 ноября

Цель работы: изучение способа создания новых процессов.

В базе данных, написанной на Lazarus/Delphi, одну из долговыполняемых операций реализуйте так, чтобы она выполнялась в отдельном потоке.

Темы для изучения: TThread, критические секции

10 Практическая работа №9. Динамическое программирование (4 часа)

Срок сдачи: 30 ноября

Цель работы: изучить методику решения задачи динамического программирования на простейшем примере.

1. Найти файл (выведите на экран его имя), имеющий наибольшую длину среди файлов, расположенных в текущем каталоге или его подкаталоге любой степени вложенности. Если таких файлов несколько, то найдите любой из них.

2. Найти файл (выведите на экран его имя), имеющий наименьшую длину среди файлов, расположенных в текущем каталоге или его подкаталоге любой степени вложенности. Если таких файлов несколько, то найдите любой из них.

3. Найти файл (выведите на экран его имя), имеющий наибольшую длину имени среди файлов, расположенных в текущем каталоге или его подкаталоге любой степени вложенности. Если таких файлов несколько, то найдите любой из них.

4. Найти файл (выведите на экран его имя), имеющий наименьшую длину имени среди файлов, расположенных в текущем каталоге или его подкаталоге любой степени вложенности. Если таких файлов несколько, то найдите любой из них.

5. Найти файл (выведите на экран его имя), имеющий наиболее позднее время создания в текущем каталоге или его подкаталоге любой степени вложенности. Если таких файлов несколько, то найдите любой из них.

6. Найти файл (выведите на экран его имя), имеющий наиболее раннее время создания в текущем каталоге или его подкаталоге любой степени вложенности. Если таких файлов несколько, то найдите любой из них.

7. Найти количество файлов, имеющих размер больше 1Кб в текущем каталоге и его подкаталогах любой степени вложенности.

8. Найти количество файлов, имеющих время создания более года назад в текущем каталоге и его подкаталогах любой степени вложенности.

9. Найти каталог (выведите его имя на экран) в текущем каталоге и его подкаталогах любой степени вложенности, имеющий наибольшее число файлов, оканчивающихся на знак `~`. Если таких каталогов несколько, то выведите любой из них.

10. Найти каталог (выведите его имя на экран) в текущем каталоге и его подкаталогах любой степени вложенности, имеющий наибольшее число файлов, имеющих нулевой размер. Если таких каталогов несколько, то выведите любой из них.

11. Найти каталог (выведите его имя на экран) в текущем каталоге и его подкаталогах любой степени вложенности, имеющий наибольшее число файлов, имеющих расширение `.txt`. Если таких каталогов несколько, то выведите любой из них.

12. Найти каталог (выведите его имя на экран) в текущем каталоге и его подкаталогах любой степени вложенности, имеющий суммарный наибольший размер файлов. Если таких каталогов несколько, то выведите любой из них.

13. Найти каталог (выведите его имя на экран) в текущем каталоге и его подкаталогах любой степени вложенности, имеющий суммарный наименьший размер файлов. Если таких каталогов несколько, то выведите любой из них.

14. Найти каталог (выведите его имя на экран) в текущем каталоге и его подкаталогах любой степени вложенности, имеющий наибольшее число подкаталогов. Если таких каталогов несколько, то выведите любой из них.

15. Найти каталог (выведите его имя на экран) в текущем каталоге и его подкаталогах любой степени вложенности, имеющий наименьшее число подкаталогов. Если таких каталогов несколько, то выведите любой из них.

16. Найти каталог (выведите его имя на экран) в текущем каталоге и его подкаталогах любой степени вложенности, имеющий наибольшее число файлов. Если таких каталогов несколько, то выведите любой из них.

17. Найти каталог (выведите его имя на экран) в текущем каталоге и его подкаталогах любой степени вложенности, имеющий наименьшее число файлов. Если таких каталогов несколько, то выведите любой из них.

18. Найти каталог (выведите на экран его имя), имеющий наиболее позднее время создания в текущем каталоге или его подкаталоге любой степени вложенности. Если таких файлов несколько, то найдите любой из них.

19. Найти каталог (выведите на экран его имя), имеющий наиболее раннее время создания в текущем каталоге или его подкаталоге любой степени вложенности. Если таких файлов несколько, то найдите любой из них.

20. Найти каталог (выведите его имя на экран) в текущем каталоге и его подкаталогах любой степени вложенности, имеющий наибольшее число файлов, доступных для чтения (см. `S_IRUSR`). Если таких каталогов несколько, то выведите любой из них.

21. Найти каталог (выведите его имя на экран) в текущем каталоге и его подкаталогах любой степени вложенности, имеющий наименьшее число файлов, доступных для чтения (см. `S_IRUSR`). Если таких каталогов несколько, то выведите любой из них.

22. Найти файл (выведите на экран его имя), имеющий наибольшее расширение среди файлов, расположенных в текущем каталоге или его подкаталоге любой степени вложенности. Если таких файлов несколько, то найдите любой из них.

23. Найти файл (выведите на экран его имя), имеющий наименьшее расширение среди файлов среди имеющих расширение, расположенный в текущем каталоге или его подкаталоге любой степени вложенности. Если таких файлов несколько, то найдите любой из них.

24. Найти файл (выведите на экран его имя), имеющий наибольшее число цифр в имени, среди расположенных в текущем каталоге или его подкаталоге любой степени вложенности. Если таких файлов несколько, то найдите любой из них.

25. Найти файл (выведите на экран его имя), имеющий наименьшее ненулевое число цифр в имени, среди расположенных в текущем каталоге или его подкаталоге любой степени вложенности. Если таких файлов несколько, то найдите любой из них.

26. Найти пустой каталог (выведите на экран его имя), имеющий наибольшую степень вложенности, среди расположенных в текущем каталоге или его подкаталоге любой степени вложенности. Если таких каталогов несколько, то найдите любой из них.

11 Практическая работа №10. Разделяемая память и семафоры в Linux (12 часов)

Срок сдачи: 21 декабря

Цель работы: изучение семафоров и разделяемой памяти

При выполнении задания используются следующие функции: `sem_open`, `sem_close`, `sem_post`, `sem_wait`, `sem_unlink`; `shm_open`, `ftruncate`, `mmap`, `munmap`, `shm_unlink`, `close`.

Решите практическую работу №9 так, чтобы каждый каталог обрабатывался отдельным процессом, а межпроцессное взаимодействие происходило с использованием разделяемой памяти и регулировалось бы семафорами.

12 Практическая работа №11. Разделяемая память и критические секции в Windows (4 часа)

Срок сдачи: 28 декабря

Цель работы: изучение критических секций

При выполнении задания используются следующие функции: InitializeCriticalSection, EnterCriticalSection, LeaveCriticalSection, DeleteCriticalSection.

Решите практическую работу №9 в среде Windows так, чтобы каждый каталог обрабатывался отдельным потоком, используя критические секции для регулирование межпотокового взаимодействия.

13 Практическая работа №12. Передача сообщений в Linux (4 часа)

Срок сдачи: зачетное занятие

Цель работы: изучение потоков сообщений.

При выполнении задания используются следующие функции: `mq_open`, `mq_receive`, `mq_send`, `mq_unlink`.

Решите практическую работу №10, используя для межпроцессного взаимодействия каналы (pipe).

14 Практическая работа №13. Использование каналов в Linux (4 часа)

Срок сдачи: зачетное занятие

Цель работы: изучение каналов (pipe).

При выполнении задания используются следующие функции: pipe.

Решите практическую работу №10, используя для межпроцессного взаимодействия каналы (pipe).