

## **Лабораторная работа № 21 по языкам и методам программирования:**

### **8 факультет, I курс, II семестр 2011/12 учебного года**

#### **Программирование на интерпретируемых командных языках**

*Составить программу выполнения заданных действий над файлами на одном из интерпретируемых командных языков ОС UNIX (Shell, Cshell, Bash, ...), согласованном с преподавателем:*

1. Копирование всех файлов, для которых установлена защита от записи от всех пользователей. Имена копий генерировать путем добавления указанного префикса. Файлы-копии должны быть доступны для записи всем пользователям.
2. Объединение в один файл всех файлов с указанным суффиксом, имеющих размер меньше заданного.
3. Удаление всех файлов с указанным суффиксом, к которым не было обращений указанное число дней.
4. Переименование всех файлов с указанным суффиксом путем добавления к ним заданного префикса.
- 5.\* Реализовать *шелловскую* версию команды split на соответствующем интерпретируемом командном языке.
6. Кодирование по Цезарю всех текстовых файлов в указанном каталоге и его поддиректориях с помощью утилиты tr.
7. Удаление всех файлов с указанным суффиксом при наличии файлов с теми же именами, но с другим заданным суффиксом.
8. Создание синонима всех файлов с указанным суффиксом и числом связей, большим 1, путем перестановки суффиксов в именах исходных файлов в начало и удаления точки.
9. Удлинение имен файлов до указанной длины заданным знаком.
10. Сравнение указанного файла со всеми другими файлами, имеющими такой же суффикс, с удалением совпадающих файлов.
11. Конкатенация указанного файла заданное число раз так, чтобы при этом длина результирующего файла в байтах не превысила заданного числа.
12. Генерация заданного числа копий указанного файла. Имена копий генерировать добавлением к имени исходного файла следующей по порядку буквы или цифры, начиная с заданной буквы или цифры.
13. Конкатенация файлов наименьшей длины до тех пор, пока размер результирующего файла не превысит размера блока заданного устройства (файловой системы).
14. Удаление файлов с указанным префиксом, длина которых находится в указанном диапазоне.
15. Удлинение файлов с указанным суффиксом до заданной длины путем присоединения к ним необходимого числа копий некоторого файла.
16. Удаление файлов с указанным префиксом и с максимальными размерами, пока общий размер остающихся файлов не станет меньше заданного числа.
17. Рекурсивный обход указанного каталога и перекодировка всех *текстовых* файлов в нем и в подкаталогах.
18. Рекурсивный обход указанного каталога и замена всех разделителей пути \ на / в файлах с именем Makefile\*.
19. Подсчет числа строк в исходных файлах (\*.h, \*.cpp) директории и подкаталогов с вычислением суммарной длины (в строках).
20. Рекурсивный обход указанного каталога и вывод на экран полных путей поддиректорий.
- 21.\* Удаление всех синонимов указанного файла из указанного каталога и его поддиректорий.
22. Замена для всех файлов, имеющих размер меньше заданного, суффиксов имен на первые литеры имен файлов.
23. Запись имен и размеров всех выполняемых файлов с указанным суффиксом в заданный файл, размер которого не должен превышать заданной величины.
24. Запись имен и размеров всех невыполняемых файлов с указанным суффиксом в заданный файл, размер которых кратен размеру блока.
25. Удаление файлов, суммарная длина которых не меньше заданного числа блоков.

26. Рекурсивный обход и подсчет числа выполнимых файлов в каталоге и в поддиректориях.
27. Рекурсивный обход, подсчет числа различных суффиксов файлов в указанном каталоге и вывод уникальных суффиксов.
- 28-29.\*\*\* Интерпретация машины Тьюринга в четвёрках или в пятёрках.

*Разработанная программа должна удовлетворять не менее чем двум из следующих дополнительных условий, согласованным с преподавателем:*

1. Если параметры опущены, то их значения должны быть запрошены у пользователя.
2. Программа должна в случае ошибки в задании параметров выводить комментированный синтаксис правильного использования.
3. Если указан параметр “?”, то выводится подсказка (спецификация программы).
4. Программа должна обеспечивать возможность установки режима трассировки.
5. Программа должна обеспечивать возможность установки режима подтверждения.
6. Программа должна обеспечивать возможность исправления значений параметров после запуска в диалоговом режиме.
7. Программа должна обеспечивать позиционный способ задания параметров (список слов через пробел или запятую).
8. Если параметры опущены, то устанавливаются некоторые стандартные значения параметров.
9. Программа должна обеспечивать задание списка ключевых параметров в формате *имя=значение* через запятую.
10. Задание параметров в конфигурационном файле.

*Обязательным является условие, соответствующее номеру группы студента.*

Для тестирования программы следует создать вспомогательные командные файлы генерации необходимых множеств файлов.

Вопросы к зачёту VIII по лабораторной работе № 21.

## **ОС UNIX. ПРОГРАММИРОВАНИЕ НА ИКЯ.**

1. Понятие о программировании на ИКЯ (Shell, Cshell, bash, perl, ...).
2. Командные файлы и процедуры.
3. Интерпретаторы Shell, Cshell, bash, kshell, zshell, ...).
4. ИКЯ общего назначения (REXX, perl, ...).
5. \*Краткая характеристика Microsoft PowerShell. Пример командного файла.

Вопросы 6-12: основные конструкции и приемы программирования даются на одном из ИКЯ, по выбору преподавателя в группе.

6. Переменные. Присваивание. Понятие окружения.
7. Использование параметров в командных файлах и процедурах.
8. Интерпретация команд. Подстановка переменных и команд. Встроенные документы.
9. Вычисление выражений.
10. Проверки и ветвления.
11. Циклы.
12. Оператор выбора.

*Задания подготовлены доц. Чернышовым Л.Н., проф. Зайцевым В.Е., ст. преп Лебедевым А.В. и доц. Сошниковым Д.В.*