

Задание на большое задание по  
дисциплине  
«Функциональное, процессное и  
объектно-ориентированное  
моделирование информационных  
систем»

Мокряков А. В.,  
кандидат физико-математических наук,  
зав. каф. Искусственного интеллекта,  
прикладной математики и программирования

22 ноября 2024 г.

## Содержание

<b>1</b>	<b>Требования к ПО</b>	<b>2</b>
<b>2</b>	<b>Требования к пояснительной записке</b>	<b>3</b>
<b>3</b>	<b>Вычислительная система</b>	<b>4</b>

# 1 Требования к ПО

Разрабатываемое в рамках курсовой по дисциплине «Фундаментальная информатика» программное обеспечение (далее ПО) должно выполнять ряд требований:

- А. Должны быть выполнены требования для выбранной темы.
- Б. Работа программы начинается с вывода на экран имени автора и информации о возможностях приложения.
- В. Любой ввод в программе должен предваряться запросом.
- Г. Код приложения должен иметь правильное форматирование:
  - 1. один оператор-одна строка;
  - 2. фигурные скобки, окаймляющие блок расположены на отдельной строке и на одном уровне с заголовком блока;
  - 3. операторы внутри блока расположены с большим отступом, чем заголовок блока, и на одном уровне;
  - 4. названия модулей, функций и переменных (кроме названий счётчиков) должны иметь содержательные имена;
  - 5. в заголовочном файле могут располагаться только макроопределения, собственные типы данных и заголовки функций;
  - 6. заголовок функции и её определение должны сопровождаться комментарием, описывающим её действие.

## 2 Требования к пояснительной записке

Пояснительная записка к разрабатываемому ПО должна содержать следующую информацию

- А. Описать функциональные возможности приложения.
- Б. Описать входные данные, принимаемые приложением:
  - 1. команды, данные, вводимые пользователем;
  - 2. файлы (структуру), считываемые приложением.
- В. Описать интерфейс приложения и взаимодействие с пользователем.
- Г. Привести пример корректной работы приложения (со скриншотами).
- Д. Описать реакцию приложения на аномалии входных данных.
- Е. Перечислить разработанные модули и функции.
- Ж. Описать последовательность вызовов функций.
- З. Описать работу каждой функции:
  - 1. входные и выходные параметры;
  - 2. схему работы (блок-схема, псевдокод или словесное описание).

### 3 Вычислительная система

В рамках разрабатываемого программного обеспечения (ПО) необходимо подготовить полноценную вычислительную систему (далее ВС) с нижеследующими требованиями.

**А.** Разработать **Makefile** для выполнения компиляции, настройки и установки ВС. Обязательно должны быть следующие цели:

1. **compile** (цель по умолчанию) — обеспечивает компиляцию всего проекта. Зависит от ряда файлов ВС.
2. **user\_man** — компилирует инструкцию пользователя.
3. **in\_instruct** — компилирует инструкцию по установке.
4. **an\_instruct** — компилирует инструкцию по инструменту анализа работы ВС.
5. **license** — компилирует лицензию о распространении ВС.
6. **docs** — зависит от **user\_man**, **in\_instruct** и **license**.
7. **configure** — производится запуск настроечного скрипта и формирует файл-инструкцию **install.conf** по установке. Зависит от некоторых файлов ВС.
8. **install** — устанавливает собранную ВС в выбранные каталоги, согласно **install.conf**, также выполняет другие необходимые действия. Так же возможен запуск специального установочного скрипта. Зависит от **configure**, **docs** и некоторых файлов ВС.
9. **all** — Зависит от целей компиляции, настройки и установки.
10. **clear\_tmp** — очистка временных файлов образованных при компиляции.
11. **clear\_config** — удаляет конфигурационный файл установки.

12. `clear_compile` — удаляет скомпилированные файлы ВС и зависит от `clear_tmp`.
13. `clear` — зависит от `clear_tmp` и `clear_config`.
14. `uninstall` — для установленной ВС выполняется удаление всех установленных файлов, кроме файлов сохранений.
15. `remove` — зависит от `uninstall` и `.log` файлами.
16. `remove_all` — зависит от `remove` и удаляет каталоги с сохранениями.

#### Б. Компиляция:

1. Компиляция производится из разных модулей, на которые ПО делится по функциональному признаку.
2. Выделяется минимум один модуль, выступающий в качестве библиотеки используемых функций, реализующий основной функционал предоставляемый ПО.
3. Выделяется минимум один модуль пользовательского интерфейса.

#### В. Настройка:

1. Настройка системы производится запуском конфигурационного скрипта, через `Makefile` или непосредственно через скрипт `./configure`.
2. При настройке запрашивается у пользователя его ФИО, E-mail (проверяется), номер телефона (проверяется). На основе полученных данных формируется файл: сертификат на свободное использование указанным пользователем данной ВС. В нем также указана лицензия распространения и дата установки.
3. Также при настройке запрашивается пути установки для сертификата и самого ПО, его временных файлов, документации и файлов сохранения по умолчанию.

4. Для вышеуказанных путей возможен выбор по умолчанию, в котором должны быть предложены стандартные пути размещения файлов: для сертификата можно предложить каталог `/usr/etc/`<<Name>>, для временных файлов каталог `/tmp/`<<Name>>, для файлов сохранений и загрузки лучше создать директорию в домашнем каталоге пользователя.

#### Г. Установка:

1. Сертификат проверяется на правильность в том числе на присутствие всех параметров и их корректность. Также проверяется время создания сертификата время внутри сертификата. При прохождении проверки можно переходить к установке.
2. Хозяином сертификата устанавливается `root`. Права для сертификата устанавливаются на `0644`.
3. Права на файлы ВС устанавливаются согласно логике использования системы.
4. Согласно настройке расположить все файлы ВС в файловой системе пользователя.
5. Настроечные файлы, в том числе с путям установки, необходимо расположить в `/etc/`<<Name>>.
6. Создать каталог для `.log` файлов: `/var/log/`<<Name>>.
7. На исполняемый файл программы, должна быть сделана символическая ссылка в `/usr/bin`.
8. Также должны быть ссылки из основного каталога ПО на директорию с файлами сохранений и другие директории ВС.

#### Д. Обязательные аргументы командной строки ПО:

- с — запуск с указанным конфигурационным файлом.

- d — в `.log` файл записывается в том числе весь диалог пользователя с ПО.
- f — передача файла, как файла с которым будет работать ПО (допускается модификация в рамках логики ВС).
- h — вывод информации об аргументах командной строки и способе запуска ПО.
- i — вывод информации об авторе, студенческой группе автора, принимающем преподавателе, кафедре и ВУЗе.
- m — запуск в форме интерактивного меню функционала системы.
- o — имя файла для сохранения результата ПО по умолчанию.
- s — тихая работа ПО: `.log` файлы не создаются.
- t — не удаляет временные файлы работы ПО.

#### Е. Поведение ВС в отношении файловой системы:

1. Временные файлы (кроме `.log` файлов) должны удаляться после легального завершения программы, если не указана специальная опция при запуске.
2. `.log` файлы должны отличаться по дате и времени запуска ВС.

#### Ж. Система анализа:

1. В разрабатываемую ВС должен входить инструмент анализа `.log` файлов.
2. `.log` файлы хранят в себе по минимуму следующую информацию:
  - дата и время запуска приложения,
  - общую продолжительность работы,
  - код завершения программы, в том числе ошибки,

- ID пользователя, который запускал приложение,
  - краткие результаты работы программы: для игр — партии, их прохождение, результат, для утилит — список обработанных файлов и результаты их обработки.
3. При анализе выводится количество запусков ВС.
  4. Составляется список ошибок произошедших в процессе работы ВС.
  5. Показывается разбивка запусков по дням недели.
  6. Показывается разбивка запусков по часам.
  7. Для утилит показывается общее количество обработанных файлов.
  8. Для игр показывается общие результаты прошедших партий, приводится общий счёт.
  9. При анализе можно указывать временной диапазон для анализа `.log` файлов.
  10. Возможен анализ отдельных `.log` файлов.
  11. Основной язык для исполнения запросов — `awk`.

### 3. Состав пояснительной записки:

1. Должен быть раздел с текстовым файлом `AUTHOR` с информацией об авторе, студенческой группе автора, принимающем преподавателе, кафедре и ВУЗе.
2. Должен быть раздел с текстовым файлом `README` с описанием информации о ВС.
3. Должен быть раздел с текстовым файлом `INSTALL` с описанием инструкции об установке ВС.
4. Должен быть раздел с файлом-лицензией на распространение ВС в двух форматах: текстовый файл `LICENSE` и файл в формате `LATEX`.



5. Должен быть раздел с подготовленной инструкцией по установке в формате L<sup>A</sup>T<sub>E</sub>X. В том числе со скриншотами.
6. Должен быть раздел с руководством пользователя в формате L<sup>A</sup>T<sub>E</sub>X. В том числе со скриншотами.
7. Должен быть раздел с инструкцией по работе с инструментом анализа в формате L<sup>A</sup>T<sub>E</sub>X.
8. В курсовой должны быть представлены все вышеуказанные разделы. Кроме того должны быть приложены все файлы по настройке и установке ВС.
9. К пояснительной записке в качестве приложения, прилагается подготовленная презентация о работе ВС в формате L<sup>A</sup>T<sub>E</sub>X.