

СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ: КОМПЛЕКС ПОДГОТОВКИ ИНЖЕНЕРОВ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Бродецкая И.С.

Яшин К.Д. – доцент, к.т.н.

Целью данной работы является разработка учебного комплекса по системному программированию для подготовки инженеров. Целью разработки учебного комплекса является изучение студентами основ функционирования и принципов построения системного программного обеспечения, овладение навыками программирования на языках высокого уровня с использованием функций операционной системы и на языке ассемблера.

Разрабатываемый комплекс по системному программированию разделен на теоретическую и практическую часть. Теоретическая часть, в свою очередь, разделена на 5 основных глав, в которых дана вся необходимая информация, для выполнения практических заданий.

Глава 1. Средства реализации программных систем. В данной главе будут рассмотрены следующие основные вопросы: архитектура и логическое устройство ЭВМ с точки зрения программиста, характеристика памяти и схем адресации, форматы данных и машинных инструкций, система прерываний, система команд.

Глава 2. Ассемблер в системном программировании. В данной главе будут рассмотрены следующие основные вопросы: Понятие ассемблера. Команды и директивы языка ассемблера. Ассемблирование, редактирование связей и выполнение программ. Макросредства языка ассемблера. Макрокоманды и макроопределения. Способы построения подпрограмм. Рекурсивный вызов подпрограмм. Связь с примитивами операционной системы.

Глава 3. Язык системного программирования С. В главе рассмотрены следующие вопросы: характеристика языка С и областей его применения. Основные операционные объекты языка С. Описание данных и указателей. Выражения и операции, приоритеты выполнения операций. Операторы присваивания, условные операторы, оператор-переключатель, операторы цикла, управляющие операторы. Функции. Препроцессор языка С. Способы модульного программирования. Характеристика программ на основе функций и макрокоманд. Модули-файлы и модули-функции в языке С. Обзор структуры библиотек стандартных функций.

Глава 4. Технология разработки программных систем. В главе рассмотрены следующие вопросы: методы структурного программирования. Принцип пошаговой детализации. Уровни описания и реализации структур данных. Абстрактные структуры данных. Множества, строки, стеки, очереди, деревья, списки, графы, матрицы, таблицы. Примеры представления структур данных на языке С. Модульный принцип построения программных систем. Типы и свойства модулей. Способы передачи информации между программными модулями. Методы комплексирования программ.

Глава 5. Примеры решения задач системного программирования

Характеристика вычислительных сред. Уровни доступа к системной информации и управления ресурсами. Анализ схемы распределения памяти. Установленные драйверы и администраторы устройств. Получение информации о системных ресурсах. Схема стандартного распределения памяти, регистры состояния оборудования, порты обмена. Схема обработки прерываний. Базовые понятия многозадачных вычислительных сред.

В результате работы с комплексом по системному программированию студенты должны:

а) знать: концептуальные основы операционных систем; принципы построения системного и прикладного программного обеспечения; основы программирования с использованием системных функций; язык ассемблера современных процессоров; структуру и особенности программирования математического сопроцессора и модуля мультимедиа-расширений; б) уметь: пользоваться основными возможностями Windows 32, программировать с помощью языка ассемблера; в) иметь представление о современном уровне развития компьютерной техники и её программного обеспечения.



Рис. 1 - Классификация-системного программного обеспечения компьютера

Список использованных источников:

1. Таненбаум Э. Современные операционные системы, 2-е изд.: Пер. с англ. - СПб: Питер, 2003. - 1040 с.: ил.
2. Гордеев А. В., Молчанов А. Ю. Системное программное обеспечение: Учебник для вузов - СПб: Питер, 2003. - 736 с.: ил.
3. Вильямс А. Системное программирование в Windows 2000 для профессионалов - СПб: Питер, 2003. - 624 с.: ил.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Гумбор С. В.

Щербина Н.В. - ассистент

Целью данной работы является проведения этапа юзабилити, так как многие компании, разрабатывающие ПО, проводят самостоятельно оценку удобства использования, в большинстве случаев это не приносит должных результатов из-за того, что целевая аудитория продукта может очень сильно отличаться от разработчиков и тестировщиков по уровню квалификации (технически и в прикладной области), по решаемым задачам, по привычным способам работы со схожими продуктами.

Испытание многих продуктов пользователю предлагают в «лабораторных» условиях решить основные задачи, для выполнения которых этот продукт разрабатывался, и просят высказывать во время выполнения этих тестов свои замечания.

Процесс тестирования фиксируется в протоколе (логе) и/или на аудио- и видеоустройства — с целью последующего более детального анализа.

Если проверка эргономичности выявляет какие-либо трудности (например, сложности в понимании инструкций, выполнении действий или интерпретации ответов системы), то разработчики должны доработать продукт и повторить тестирование.

Наблюдение за тем, как люди взаимодействуют с продуктом, нередко позволяет найти для него более оптимальные решения. Если при тестировании используется модератор, то его задача — держать респондента сфокусированным на задачах (но при этом не „помогать“ ему решать эти задачи).



Рис. 1 – Пример юзабилити-теста

Основную трудность после проведения процедуры проверки эргономичности нередко представляют большие объемы и беспорядочность полученных данных. Поэтому для последующего анализа важно зафиксировать:

1. Речь модератора и респондента;
2. Выражение лица респондента (снимается на видеокамеру);
3. Изображение экрана компьютера, с которым работает респондент;
4. Различные события, происходящие на компьютере, связанные с действиями пользователя;

Все эти потоки данных должны быть синхронизированы по тайм-кодам, чтобы при анализе их можно было бы соотносить между собой.

Наряду с модератором в тестировании нередко участвуют наблюдатели. По мере обнаружения проблем они делают свои заметки о ходе тестирования так, чтобы после можно было синхронизировать их с основной записью. В итоге каждый значимый фрагмент записи теста оказывается прокомментирован в заметках наблюдателя. В идеале ведущий (т.е. модератор) представляет разработчика, наблюдатели — заказчика (например издателя, дистрибьютора), а испытуемые — конечного пользователя (например покупателя).

Кроме вышеизложенного существует еще один подход к проверке эргономичности: для решения задачи, предложенной пользователю, разрабатывается "идеальный" сценарий решения этой задачи. Как правило, это сценарий, на который ориентировался разработчик. При выполнении задачи пользователями регистрируются их отклонения от задуманного сценария для последующего анализа. После нескольких итераций доработки сайта и последующего тестирования можно получить интерфейс, удовлетворительный с точки зрения пользователя.

Список использованных источников:

1. Шупейко, И. Г. Теория и практика инженерно-психологического проектирования и экспертизы: учебно-методическое пособие к практическим видам занятий / И. Г. Шупейко. — Минск: БГУИР, 2009. — 126 с.
2. Якоб Нильсен, Хоа Лоранжер Web-дизайн: удобство использования Web-сайтов. — М.: «Вильямс», 2007.

ОНЛАЙН-МАГАЗИН ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь