**ПЛАН ЗАНЯТИЯ**

**Дисциплина:** МДК.01.04 Системное программирование

**Преподаватель:** Галузин А.Б.

**Курс:** 4

**Группа:** п-40

**Специальность:** 09.02.07 Информационные системы и программирование

**Дата:** 2.09.24

**Время проведения:** 1 пара

**Тема:** Понятие системного программирования. Состав и основные функции системного ПО.

**Цель занятия:**

**дидактическая:** сформировать понятие системного программирования; изучить состав и основные функции системного ПО.

**развивающая**: развивать абстрактное мышление, логику

**Вид занятия** лекция

**Литература**

Головнин О. К. Введение в системное программирование и основы жизненного цикла системных программ: учебное пособие – Самара: Изд-во Самарского университета, 2021, стр. 8-11.

**Интернет-ресурсы:**

<https://studme.org/94349/informatika/podsistemy_upravleniya_resursami_kompyutera>

<https://studme.org/94350/informatika/podsistema_upravleniya_pamyatyu>

<https://studme.org/94351/informatika/podsistema_upravleniya_faylami>

<https://spravochnick.ru/programmirovanie/sistemnoe_programmirovanie/>

**ЗАДАНИЕ**: законспектировать лекцию с учетом контрольных вопросов.

**КОНСПЕКТ ЛЕКЦИИ**

**Тема: Понятие системного программирования**

План

1. Введение в системное программирование
2. Системное программное обеспечение
3. Подсистемы управления ресурсами.

**1. Введение в системное программирование**

Изучение системного программирования является неотъемлемой частью подготовки специалистов в области информационных технологий и вычислительной техники.

Целью освоения междисциплинарного курса (МДК), входящего в состав профессионального модуля, является формирование теоретических знаний и практических умений в области системного программирования.

Задачи дисциплины:

* ознакомление с основами функционирования и устройства системного программного обеспечения (СПО);
* приобретение практических навыков программирования с использованием языка низкого уровня;
* приобретение навыков работы с языками программирования для создания СПО;
* получение практических навыков использования современных инструментальных средств, стандартных библиотек классов и шаблонов;
* формирование знаний и умений разработки СПО на объектно-ориентированных языках программирования;
* получение практических навыков по разработке и оформлению проектной и рабочей технической документации на СПО

Рассмотрим классификацию программного обеспечения. На сегодняшний день выделяют по меньшей мере три класса программного обеспечения (ПО): прикладное, промежуточное и системное.

**Прикладное ПО** – компьютерные программы, предназначенные для решения функциональных задач в определенной предметной области.

Это самый многочисленный класс программных продуктов.

Примеры прикладного ПО: текстовые редакторы (Microsoft Word, Libre Office), табличные редакторы (Microsoft Excel), графические редакторы (Adobe Photoshop), системы управления базами данных (СУБД);

**Промежуточное (связующее программное обеспечение)** (англ. *middleware*; *программное обеспечение среднего слоя*, *подпрограммное обеспечение*, *межплатформенное программное обеспечение*) — широко используемый термин, означающий слой или комплекс технологического программного обеспечения для обеспечения взаимодействия между различными приложениями, системами, компонентами (мониторы транзакций, система распределённых объектов (DOS, англ. distributed object system) или технологии распределённых объектов (DOT, англ. distributed object technology), вызов удалённых процедур (RPC, англ. remote procedure call), ориентированное на обработку сообщений (MOM, англ. message-oriented middleware) .

**Системное ПО** – совокупность программ, предназначенных для поддержания работоспособности системы обработки информации и управления (компьютера и компьютерных сетей) или повышения эффективности ее использования.

Функции системного ПО:

* создание операционной среды функционирования для программ;
* автоматизация разработки новых программ;
* обеспечение надежной и эффективной работы компьютера и компьютерной сети;
* проведение диагностики и профилактики аппаратуры
* компьютера и компьютерных сетей;
* выполнения вспомогательных технологических процессов (копирование, архивирование, восстановление после сбоев и т.п.).

Системное ПО тесно связано с типом компьютера и является его неотъемлемой частью. Системное ПО ориентировано на квалифицированных пользователей и профессионалов в области информационных технологий.

К системному ПО предъявляются высокие требования по надежности и технологичности работы, удобству и эффективности использования.

Существуют следующие группы системного ПО:

* операционные системы (ОС);
* интерфейсные оболочки ОС;
* системы управления файлами;
* системы программирования;
* утилиты;
* драйверы;
* средства сетевого доступа.

Системное программирование – это процесс разработки системных программ.

Другой взгляд на системное программирование – разработка программ сложной структуры.

Значительная часть системного и практически все прикладное ПО пишется на языках высокого уровня, что обеспечивает сокращение расходов на их разработку, модификацию и переносимость.

Однако, разработка системного ПО предполагает знание и использование ассемблера для создания модулей и ассемблерных вставок.

**2. Системное программное обеспечение**

Существуют несколько способов классификации системного ПО. Ниже рассматриваются основные.

Традиционная классификация системного ПО:

* управляющее ПО – организует корректное функционирование всех процессов ОС и устройств компьютера;
* обрабатывающее ПО – обеспечивает выполнение специальных задач.

Управляющее системное ПО осуществляет оркестрирование вычислительных процессов, организует работу с внутренними данными ОС. Этот вид ПО располагается в основной памяти и представляет собой резидентные программы ОС. Системное управляющее ПО поставляется в виде инсталляционных пакетов ОС и драйверов устройств.

Обрабатывающее системное ПО функционирует в операционной среде ОС с целью выполнению специальных задач. Системное обрабатывающее ПО поставляется в виде дистрибутивных пакетов с программами инсталляции, разворачивающими ПО на конечном устройстве.

Альтернативная классификация системного ПО:

* базовое ПО – минимальный набор программных средств, обеспечивающих работу компьютера и компьютерной сети;
* сервисное ПО – программы и программные комплексы, которые расширяют возможности базового ПО и организуют удобную среду для работы других программ и пользователя.

Примеры базового ПО:

* операционные системы и драйверы в составе ОС;
* загрузчики;
* интерфейсные оболочки ОС;
* системы управления файлами.

Операционная система – совокупность программных средств, обеспечивающая управление аппаратной частью компьютера и прикладными программами, а также их взаимодействием между собой и пользователем.

Загрузчик – программа, обеспечивающая выбор ОС при запуске компьютера.

Интерфейсные оболочки ОС предназначены для смены режима взаимодействия пользователя с прикладными программами и ресурсами компьютера. Они могут функционировать в текстовом или графическом режимах. Примеры интерфейсных оболочек: KDE, Gnome.

Системы управления файлами выделяются в отдельную группу системного ПО, однако любая система управления файлами разработана для работы в конкретной ОС и с конкретной файловой системой. Системы управления файлами предназначены для организации более удобного доступа к данным, когда вместо низкоуровневого доступа к данным с указанием конкретных физических адресов используется логический доступ с указанием имени файла.

Примеры сервисного ПО:

* драйверы специальных устройств, которые поставляются отдельно от ОС;
* программы диагностики работоспособности компьютера;
* антивирусные программы;

Сервисное ПО часто называются утилитами. Утилиты – программы, служащие для выполнения вспомогательных операций обработки данных или обслуживания компьютеров. Примеры утилит: CCleaner, 7-Zip, WinRAR, AIDA, Disk Commander.

**3. Подсистемы управления ресурсами**

*Подсистемой* называют группу функций, обеспечиваемых ОС. Наиболее важными подсистемами управления ресурсами являются подсистемы управления процессами, памятью, файлами и внешними устройствами. Подсистемами, общими для всех ресурсов, являются подсистемы пользовательского интерфейса, защиты данных и администрирования.

**Подсистема управления процессами**

является важнейшей частью ОС, влияющей непосредственно на функционирование компьютера. Чтобы процесс мог быть выполнен, ОС назначает ему область оперативной памяти для размещения кодов и данных, предоставляет необходимое количество процессорного времени, а также доступ к файлам и устройствам ввода-вывода. В обязанности ОС входит поддержание очередей заявок процессов на ресурсы, поскольку процессы часто одновременно претендуют на одни и те же ресурсы, например, в виде очереди к процессору, принтеру, последовательному порту. Важной задачей ОС является защита ресурсов, выделенных данному процессу, от остальных процессов. Наиболее важными из защищаемых ресурсов процесса являются области оперативной памяти, в которых хранятся коды и данные процесса.

Выполнение процесса может быть прервано и продолжено. Для возобновления процесса необходимо восстановить состояние его операционной среды, которое идентифицируется (определяется) состоянием регистров и программного счетчика, режимом работы процессора, указателями на открытые файлы, информацией о незавершенных операциях ввода-вывода, кодами ошибок, выполняемых данным процессом системных вызовов и т.д. Эта информация называется *контекстом процесса.* При смене процесса происходит переключение контекстов.

Операционная система выполняет также функции *синхронизации* процессов, т.е. может приостанавливать выполнение процесса до наступления какого-либо события в системе, например до завершения операции ввода/вывода, осуществляемой по запросу процесса самой ОС.

Таким образом, подсистема управления процессами планирует выполнение процессов, т.е. распределяет процессорное время между несколькими одновременно существующими в системе процессами, занимается созданием и уничтожением процессов, обеспечивает процессы необходимыми системными ресурсами, поддерживает синхронизацию процессов, а также обеспечивает взаимодействие между процессами.

**Подсистема управления памятью**

осуществляет распределение имеющейся физической памяти между всеми существующими в системе в данный момент процессами, загрузку кодов и данных процессов в отведенные им области памяти, настройку адресно-зависимых частей кодов процесса на физические адреса выделенной области, а также защиту областей памяти каждого процесса.

Для управления памятью в современных ОС используется механизм *виртуальной памяти,* который состоит в отображении адресов (к которым может обратиться программа) из виртуального адресного пространства в адреса реальной памяти. Наличие в ОС механизма виртуальной памяти позволяет программисту писать программу так, как будто в его распоряжении имеется однородная оперативная память большого объема, часто существенно превышающего объем имеющейся физической памяти. В действительности все данные, используемые программой, хранятся на диске и при необходимости частями (сегментами или страницами) отображаются в физическую оперативную память. При перемещении кодов и данных между оперативной памятью и диском подсистема виртуальной памяти выполняет трансляцию *виртуальных адресов,* полученных в результате компиляции и компоновки программы, в физические адреса ячеек оперативной памяти.

Защита памяти состоит в том, чтобы предохранять выполняемую задачу от записи или чтения памяти, назначенной другой задаче. Средства защиты памяти должны пресекать несанкционированный доступ процессов к чужим областям памяти.

Таким образом, функциями ОС по управлению памятью являются отслеживание свободной и занятой памяти; выделение памяти процессам и освобождение ее при завершении процессов; защита памяти; вытеснение процессов из оперативной памяти на диск, когда размеры основной памяти недостаточны для размещения в ней всех процессов, и возвращение их в оперативную память, когда в ней освобождается место, а также настройка адресов программы на конкретную область физической памяти.

**Подсистема управления файлами**

преобразует символьные имена файлов, с которыми удобно работать пользователю или программисту, в физические адреса данных на диске, организует совместный доступ к файлам, защищает их от несанкционированного доступа. В качестве файлов могут быть использованы некоторые устройства (принтер, сетевой адаптер и др.), т.е. ОС наделена способностью работать с ними как с файлами.

**Подсистема управления внешними устройствами**

(ВУ), или устройствами ввода-вывода, обеспечивает интерфейс (сопряжение) со всеми устройствами, подключенными к компьютеру. Например, подсистема управления ВУ по запросам файловой подсистемы осуществляет передачу данных между дисками и оперативной памятью. Одной из наиболее важных задач ОС является поддержание высокоуровневого унифицированного интерфейса ввода-вывода. Эта задача решается с помощью драйверов, созданием которых занимаются как разработчики конкретной ОС, так и специалисты компаний, выпускающих ВУ. Унифицированный интерфейс ввода-вывода в большинстве ОС строится на основе концепции файлового доступа, согласно которой обмен с любым ВУ выглядит как обмен с файлом, имеющим имя и представляющим собой неструктурированную последовательность байтов.

**Подсистема защиты данных и администрирования**

Операционная система (ОС) должна обеспечить защиту от сбоев и отказов аппаратуры, ошибок программного обеспечения, ошибочного или злонамеренного поведения пользователей системы. Функции защиты ОС тесно связаны с функциями администрирования. Поддержка отказоустойчивости входит в обязанности системного администратора. Администратор определяет права пользователей при их обращении к разным ресурсам системы (файлам, каталогам, принтерам, сканерам и другим устройствам) и ограничивает их возможности в выполнении тех или иных системных действий. Поэтому для защиты данных от несанкционированного доступа ОС, прежде всего, должна убедиться в том, что в систему пытается войти пользователь, вход которого разрешен администратором.

**Вопросы для самоконтроля**

1. Что представляет собой прикладное ПО? Приведите примеры прикладного ПО.
2. Что представляет собой промежуточное ПО? Приведите примеры промежуточного ПО.
3. Что представляет собой системное ПО?
4. Какие функции выполняет системное ПО?
5. Какие группы системного ПО существуют?
6. Дайте определение системного программирования
7. На какие классы разделяют системное ПО в рамках традиционной классификации?
8. На какие классы разделяют системное ПО в рамках альтернативной классификации?
9. Приведите примеры базового ПО.
10. Дайте определение операционной системы.
11. Для чего предназначены интерфейсные оболочки ОС?
12. Что представляют собой системы управления файлами? Для чего они предназначены?
13. Приведите примеры сервисного ПО
14. Какой принцип используют в Linux для осуществляется обмена информацией с ВУ?