**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:** 3.09.24

**Время проведения:** 3 пара

**Тема:** Системы программирования и системные вспомогательные программы.

**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. 14-16, 26-28.

**Интернет-ресурсы:**

<https://studme.org/94376/informatika/sredstva_sozdaniya_prilozheniy>

<https://studme.org/94380/informatika/sredstva_programmirovaniya_otladki>

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

**КОНСПЕКТ ЛЕКЦИИ**

**Тема: Системы программирования и утилиты**

План

1. Системы программирования
2. Системные вспомогательные программы

1. **Системы программирования**

*Системы программирования* – комплексы программ и прочих средств, предназначенных для разработки и эксплуатации программ на конкретном языке программирования для конкретного вида ЭВМ.

Можно выделить два больших класса инструментальных средств:

• для создания отдельных приложений (программ);

• для создания информационных систем и технологий.

Средства для создания отдельных приложений включают:

* локальные средства (языки программирования, системы программирования, инструментальные среды пользователя)
* интегрированные среды разработки программ (IDE – от англ. Integrated Development Environment), основное назначение которых - повышение производительности труда программистов за счет автоматизации создания кодов программ, обеспечивающих интерфейс пользователя графического типа, а также автоматизации разработки запросов и отчетов (например, Delphi, Visual Studio).

В свою очередь языки программирования делятся на следующие виды:

1. операторные. Используются для кодирования алгоритмов, а потому также называются алгоритмическими. Включают:

• машинно-зависимые (ассемблер). Применяются для написания программ, явно использующих специфику конкретной аппаратуры. Каждый компьютер имеет такую систему программирования, которая изготавливается и поставляется фирмой-изготовителем вместе с компьютером;

• машинно-ориентированные (язык С). Объединяет идеи ассемблера и алгоритмического языка. Программы компактны и работают очень быстро.

• универсальные (Турбо-Паскаль, Бэйсик). Приближены максимально, насколько это возможно, к естественному английскому языку: название каждой команды – английское слово;

2. функциональные. Применяются, как правило, для машинного моделирования той или иной проблематики. Имеют в составе:

* проблемно-ориентированные (GPSS, англ. General Purpose Simulation System – общецелевая система моделирования). Моделируют систему с помощью последовательности событий. Применяются, в частности, при проектировании вычислительных комплексов (Kotlin);
* объектно-ориентированные (**Forth**). Имеют встроенные средства для моделирования новых объектов программирования;
* логико-ориентированные (Prolog). Отдельно описываются правила предметной области, по которым затем выводятся новые факты.

Системы программирования включают:

* *интегрированную среду разработчика программы*, состоящую из текстового редактора (позволяющего создавать и корректировать исходные тексты программ), средств поддержки интерфейса программиста с системными средствами для выполнения различных сервисных функций (сохранения, открытия файла…);
* *транслятор* – программу, переводящую исходный текст во внутреннее представление компьютера;
* *отладчик* – программу для трассировки и анализа выполнения прикладных программ (позволяет отслеживать выполнение программы в пооператорном режиме, идентифицировать место и вид ошибок в программе, наблюдают за изменением значений переменных, выражений и т.д.);
* *редактор связей (компоновщик)* – программа для подготовки прикладной программы к работе в конкретных адресах основной памяти компьютера;
* справочные системы.



**Технология создания загрузочного модуля**

Трансляторы разделяются на три основных класса: ассемблеры, компиляторы и интерпретаторы.

*Ассемблер* – системная программа, которая преобразует символические конструкции в команды машинного языка. Под машинным языком тут и далее понимается язык конкретного компьютера (процессора) или группы компьютеров (семейства процессоров). Ассемблер осуществляет дословную трансляцию одной символической команды в одну машинную. Ассемблеры обеспечивают повышение эффективности программирования за счет того, что программист легче запоминает и воспринимает мнемоническое обозначение машинных команд, а не их двоичный код.

*Компилятор* – системная программа, выполняющая трансляцию программы на исходном языке программирования в программу на машинном языке. В связи с тем, что команды исходного языка значительно отличаются от команд машинного языка, одна команда исходного языка может транслироваться в 5-10 и более машинных команд. Процесс трансляции с таких языков обычно называется компиляцией, а исходные языки обычно относятся к языкам программирования высокого уровня.

*Интерпретатор* – системная программа, осуществляющая пооператорную трансляцию и выполнение исходной программы. В отличие от компилятора, интерпретатор не порождает на выходе программу на машинном языке, распознав команду исходного языка, он тут же выполняет ее. Интерпретатор позволяет начать обработку данных после написания даже одной команды, что делает процесс разработки и отладки программ более гибким. Отсутствие выходного машинного кода позволяет не порождать дополнительные файлы, а сам интерпретатор можно достаточно легко адаптировать к любым машинным архитектурам. У интерпретаторов два основных недостатка: первый – относительно низкая скорость работы интерпретируемых программ, второй – исходный код программы остается доступен конечным пользователям.

Инструментальная среда пользователя – это специальные программные средства, встроенные в ППП:

* библиотеки функций, процедур, объектов и методов обработки;
* макрокоманды;
* программные модули-вставки;
* конструкторы экранных форм и отчетов;
* языки запросов высокого уровня.

Средства для создания информационных систем и технологий (CASE-технология (Computer-Aided System Engineering – автоматизированная разработка программного обеспечения) поддерживают полный цикл проектирования сложной информационной системы или технологии от исследования объекта автоматизации до оформления проектной и прочей документации на информационную систему или технологию. Они позволяют вести коллективную работу над проектом за счет возможности работы в локальной сети, экспорта – импорта любых фрагментов проекта, организации управления проектом.

**2. Системные вспомогательные программы**

К системным программам можно отнести большое количество так называемых утилит, т.е. программ вспомогательного назначения.

Утилита (английское utility или tool) — программный продукт, предназначенный не для решения какой-либо прикладной задачи, а для решения вспомогательных задач.

Разновидности утилит.

1. Программы-упаковщики (архиваторы) позволяют за счет применения специальных средств сжимать информацию на дисках, например, создавать копии файлов меньшего размера, а также объединять несколько файлов в один архивный файл. При создании архивов используются такие алгоритмы сжатия, которые позволяют уплотнить данные в 2 и более раз. Наиболее популярные программы- упаковщики — PKZIP, ARJ, RAR., WinRAR, WinZip.

2. Программы для создания резервных копий информации позволяют быстро скопировать информацию, находящуюся на жестком диске компьютера, на дискеты или кассеты стримера. Из этих программ широко используются Norton Backup, Fast Back Plus.

3. Антивирусные программы предназначены для предотвращения заражения компьютерным вирусом и ликвидации последствий заражения.

4. Программы для диагностики компьютера позволяют проверить конфигурацию компьютера (количество памяти, ее использование, типы дисков и т.д.), а также проверить работоспособность устройств компьютера. Утилиты для устранения дефектов на дисках называет дисковыми скан-корректорами. Наиболее распространены такие утилиты, как MS ScanDisk, Norton Disk Doctor из пакета NU.

5. Коммуникационные программы предназначены для организации обмена информацией между компьютерами. Программы типа Brooklin Bridge, DeskLink, Lap Link III и др. позволяют пересылать файлы с одного компьютера на другой при соединении кабелем их последовательных портов. Аналогичные возможности имеются в пакете NU. Другой вид программ (Telemate, Procomm и др.) обеспечивает возможность связи компьютеров по телефонной сети (при наличии модема). Программы типа BitFax, Fax-It и другие дают возможность посылать и принимать телефаксные сообщения при наличии в компьютере встроенной платы факс-модема.

6. Программы для оптимизации дисков позволяют обеспечить более быстрый доступ к информации на них. Эти программы перемещают все участки каждого файла друг к другу (устраняют дефрагментацию), собирают все файлы в начале диска и т.д., за счет чего уменьшается число перемещений головок диска (т.е. ускоряется доступ к данным) и снижается его износ. Программа дефрагментации имеется в Windows - Disk Defragmenter, а также в пакете NU - Speed Disk.

7. Программы динамического сжатия дисков позволяют увеличить количество информации, хранимой на дисках. Эти программы сжимают информацию при записи на диск, а при чтении — восстанавливают ее в исходном виде. Для пользователя эти программы незаметны. Возможно сжатие баз данных, например, в 4-5 раз.

8. Программы автономной печати (спулеры) позволяют распечатывать файлы на принтере параллельно с выполнением другой работы на компьютере.

Существуют утилиты, расширяющие функциональные возможности существующих прикладных или системных средств, например, утилиты автоматизации и конвертации медиа-файлов (mp3 id3, Ogg tags), безопасного (невосстановимого) удаления файлов, утилиты переконвертирования файлов различных форматов, утилиты системного менеджмента.

**Вопросы для самоконтроля**

1. Что собой представляют системы программирования? На какие два класса они делятся?
2. Что включают системы программирования? Опишите каждый компонент
3. Какие бывают трансляторы?
4. Чем отличается компилятор от интерпретатора?
5. Для чего нужен эмулятор?
6. Что такое утилита?
7. Какие разновидности утилит Вы знаете?