**Модели процессов обработки информации. Централизованная, децентрализованная и смешанная формы обработки.**

При **числовой** обработке используются такие объекты, как переменные, векторы, матрицы, многомерные массивы, константы и т.д. При **нечисловой** обработке объектами могут быть файлы, записи, поля, иерархии, сети, отношения и т.д. Другое отличие заключается в том, что при числовой обработке содержание данных не имеет большого значения, в то время как при нечисловой обработке нас интересуют непосредственные сведения об объектах, а не их совокупность в целом.

**С точки зрения реализации на основе современных достижений вычислительной техники выделяют следующие виды обработки информации:**

• последовательная обработка, применяемая в традиционной фоннеймановской архитектуре ЭВМ, располагающей одним процессором;

• параллельная обработка, применяемая при наличии нескольких процессоров в ЭВМ;

• конвейерная обработка, связанная с использованием в архитектуре ЭВМ одних и тех же ресурсов для решения разных задач, причем если эти задачи тождественны, то это последовательный конвейер, если задачи одинаковые — векторный конвейер.

Принято относить существующие архитектуры ЭВМ с точки зрения обработки информации к одному из следующих классов.

**Архитектуры с одиночным потоком команд и данных (SISD).** К этому классу относятся традиционные фоннеймановские однопроцессорные системы, где имеется центральный процессор, работающий с парами «атрибут — значение».

**Архитектуры с одиночными потоками команд и данных (SIMD).** Особенностью данного класса является наличие одного (центрального) контроллера, управляющего рядом одинаковых процессоров.

**Архитектуры с множественным потоком команд и одиночным потоком данных (MISD). К** этому классу могут быть отнесены конвейерные процессоры.

**Архитектуры с множественным потоком команд и множественным**

**потоком данных (MIMD).** К этому классу могут быть отнесены следующие конфигурации: мультипроцессорные системы, системы с мультобработкой, вычислительные системы из многих машин, вычислительные сети.

**Основные процедуры обработки данных представлены на рис.** 

       Централизованные формы применения средств вычислительной техники, которые существовали до массового использования персональных электронно-вычислительных машин (ПЭВМ), предполагали их сосредоточение в одном месте и организацию информационно-вычислительных центров (ИВЦ) индивидуального и коллективного пользования (ИВЦКП). Деятельность ИВЦ и ИВЦКП характеризовалась обработкой больших объемов информации, использованием нескольких средних и больших ЭВМ, квалификационным персоналом для обслуживания техники и разработки программного обеспечения. Централизованное  
применение вычислительных и других технических средств позволяло организовать их надежную работу, планомерную загрузку и квалификационное обслуживание.  
     При централизованных формах, когда у пользователей нет непосредственного контакта с ЭВМ, его роль сводится к передаче исходных данных на обработку, получению результатов, выявлению и устранению ошибок.

    Децентрализованные формы использования вычислительных ресурсов начали формироваться со второй половины 80-х годов, когда сфера экономики получила возможность перейти к массовому использованию персональных ЭВМ. Децентрализация предусматривает размещение ПЭВМ в местах возникновения и потребления информации, где создаются автономные пункты ее обработки. К ним относятся абонентские пункты и автоматизированные рабочие места (АРМ). **Централизованная обработка**:

***плюсы***:  
- нет накладных расходов, связанных с согласованием информации в разных местах

- полный контроль над системой в одном месте

- проще разработка

***минусы***:  
- ограничение производительности/пропускной способности - мощность одной машины может расти не бесконечно

- меньшая степень надёжности - одно звено определяет работоспособность всей системы

**Децентрализованная обработка**:

**плюсы**:  
- лучше масштабируемость - то есть можно наращивать пропускную способность, производительность увеличением кол-ва компонентов системы  
- больше надёжность - при отказе части компонентов вся система может продолжать работать, пусть и не в полной мере

**минусы**:  
- необходимость синхронизации возможно противоречивых данных из разных источников, накладные расходы на это

- сложнее разрабатывать такое ПО