

Централизованные системы.

- 1) Централизованная система может состоять из различных частей (компонентов). Однако эти части, как и классы в объектно-ориентированной программе, не являются автономными; в любой момент времени система обладает над ними полным контролем. Части системы вступают в работу в момент ее запуска, и если система отключается или удаляется, то все ее части также будут отключены или удалены.
- 2) Централизованные системы зачастую бывают довольно однородными. При их конструировании используются, как правило, одна и та же технология. Очень часто для разработки всех компонентов используется один и тот же язык программирования и компилятор. Компилятор генерирует машинный код одного типа, и во всех компонентах используется одинаковое представление данных. Затем компоненты статически или динамически загружаются одним и тем же загрузчиком или компоновщиком. Эффективное взаимодействие частей системы может быть организовано с помощью примитивов языка программирования, таких как вызовы процедур или методов.
- 3) Некоторые централизованные системы являются многопользовательскими. Хорошим примером служат приложения баз данных. База данных, ее формы и отчеты представляют собой разделяемые ресурсы. Все пользователи совместно используют одну и ту же систему, и временами она может испытывать перегрузку.
- 4) Централизованная система может быть построена так, что в ней будет выполняться единственный процесс. Ее можно сконструировать даже как однопоточную, и проектировщики могут строить централизованные системы, которые не являются одновременно выполняемыми программами.
- 5) В централизованной системе есть только одна точка управления (point of control)

Распределенные системы.

- 1) Распределенным системам присуща более высокая степень модульности. Они содержат множество компонентов, которые также могут быть разбиты на части. Эти компоненты являются автономными, в любой момент времени они имеют полный контроль над своими частями. При этом не существует главного компонента, который бы управлял всеми остальными компонентами распределенной системы, чтобы распределенная система выглядела как интегрированное вычислительное средство, для компонентов должны быть определены интерфейсы, при помощи которых они могли бы взаимодействовать друг с другом.
- 2) Компоненты распределенной системы не обязательно должны быть однородными. Неоднородность обуславливается, например, потребностью в интеграции компонентов унаследованной аппаратурой платформы, такой же мейнфрейм IBM, со вновь написанными компонентами, предназначенными для запуска на рабочей станции с операционной системой Unix или машине с Windows NT. При этом могут быть написаны на разных языках программирования: программы для мейнфрейма часто пишутся на языке ассемблера, RPC, [удаленные вызовы процедур – RPC, обеспечивают вызов процедур за пределами хостов, с RPC началась разработка объектно-ориентированного промежуточного слоя, к которому относятся различные

реализации CORBA, удаленный вызов методов (RMI) в системе JAVA, а также Microsoft COM] или Коболе, тогда как новые компоненты пишутся на языке Visual Basic, JAVA или C++. Исходный текст компонентов может быть откомпилирован в неоднородный машинный код. В этом машинном коде могут использоваться разные форматы представления данных. [Мэйнфреймы IBM используют расположение целых чисел, начиная со старших разрядов, и кодировку символов EBCDIC, тогда как большинство рабочих станций с операционной системой Unix используют расположение целых чисел, начиная с младших разрядов и 7-битную кодировку ASCII или 8-битную кодировку ISO].

- 3) Распределенная система может содержать компоненты, используемые только одним пользователем (они также известны как неразделяемые), т.е. могут использоваться в монопольном режиме. Фактически, это одно из основных достоинств распределенных систем. Если компонент перегружен слишком большим количеством пользователей или заявок от других компонентов, то в распределенную систему можно добавить другой компонент, способный предоставлять те же услуги. И разделить между ними нагрузку. Кроме того, компоненты могут размещаться так, чтобы они были локальными для пользователей и других компонентов, с которыми взаимодействуют. Эта локальность поддерживает общую производительность распределенной системы, поскольку локальные связи в общем случае более эффективны, чем связи между удаленными друг от друга компонентами.
- 4) Вследствие автономности компонентов в распределенных системах компоненты работают одновременно. В общем случае число процессов в системе равно количеству компонентов. Кроме того, компоненты часто являются многопоточными; при каждом запуске они могут создавать новый поток, обслуживающий пользователя или другой компонент. Таким образом, занятый обслуживанием компонент не блокируется и доступен для следующих заявок. В дополнение к этому процессы распределенной системы обычно выполняются не на одном процессоре. Следовательно, взаимодействие процессов включает связь по сети с другими машинами.
- 5) В отличие от централизованных систем распределенные системы имеют более одной точки отказа. Система может выйти из строя из-за отказа какого-либо компонента. Она может отказать при неисправности сети, а также когда нагрузка на компонент столь высока, что она не может откликаться в приемлемом временном интервале. Следовательно, в распределенной системе возможны такие ситуации, когда одни ее части полностью работоспособны, тогда как другие части, зависящие от недоступных в данный момент компонентов, не могут нормально функционировать.