ЛЕКЦИЯ 3

Одной из основных частей АСУ ТП является ПО, представляющее собой совокупность взаимосвязанных и автономных программ, описаний, инструкций программиста и пользователя, тестов и т.п.

**Надёжность ПО** – есть свойство программного обеспечения своевременно выполнять в заранее указанных условиях эксплуатации вперёд установленных функций.

В самом общем случае основную функцию ПО АСУТП можно рассматривать как своевременное получение некоторого результата или решения y при переработке входной информации x из множества X.

Под x понимается контрольная информация, сигналы о состоянии технологического оборудования, команды управленческого персонала и вышестоящих АСУ и т.п.

Результат y зависит как от случайного x принадлежит X, так и от свойства ПО, носящих во многом стохастический характер. Поэтому установление каких-либо диапазонов изменения y и тем более границ допустимых или разумных результатов Y оказывается в этом случае невозможным. Вследствие этого становится затруднительной строгая качественная оценка принадлежности данного y множеству «разумных» результатов Y.

**Надёжность ПО** устанавливается по результатам работы ПО, т.е. при динамической проверке всех программ на множестве входной информации. Некорректное ПО заведомо ненадёжно, одна и корректное ПО может быть ненадёжным. Рассмотренное определение надёжности ПО базируется на понятии отказа программы, под которым понимается событие, заключающееся в появлении «неразумного» результата y принадлежит Y при x принадлежит X и нормальной работе УВМ и ТСА.

Отказы ПО:

1. Неслучайные отказы обусловлены действием компьютерных вирусов.
2. Случайные отказы наблюдаются в случайные моменты времени работы УВМ или процессора. По своим последствиям эти отказы классифицируются на.
   1. Случайные сбои программ
   2. Устойчивые отказы ПО

**Под сбоем ПО** понимают случайные события, которые заключаются в появлении «неразумного» результата y «принадлежит» Y и исчезающее при последующих прогонах (запусках) программ.

**Сбой ПО –** это самоустраняющийся отказ программы, возникающий при некоторых, возможно случайных, состояниях УВМ и информации x «принадлежит» X, наблюдаемый пользователем в случайные моменты времени и исчезающий без вмешательства программиста.

**Устойчивый отказ** – наблюдается в случайный момент процессорного времени в форме «неразумного» результата y принадлежит Y при x принадлежит X в нормальном функционировании УВМ.

**Причиной отказа ПО** служит некоторая систематическая ошибка программы, после устранения которой данный отказ исчезает, т.е. имеет место восстановление ПО.

Различают ошибки **первичного** и **вторичного типа.**

**Ошибки первичного типа** связаны с неточностями в текстах программ и возникают при подготовке носителей и документации ПО, при записях кодов на алгоритмических языках и трансляции программ на машинный язык. А также из-за неточностей алгоритмов и при неверных или некорректных постановках, решаемых на УВМ.

**Ошибки вторичного типа** во многом являются следствием первичных ошибок программ. К ним относят ошибки:

1. **Вычислительные** (неверная индексация и подсчёт временных параметров, расхождение результата ручного и машинного счёта, появление неустойчивых операций и т.п.)
2. **Логические** (пропуск логических условий, неверные краевые условия и др.)
3. **Сопряжение интерфейсов** (межмодульных, программно-технических, информационных)

**Итог:** ошибки первичного и вторичного типов порождаются на этапах разработки спецификаций на ПО: проектирования ПО: реализации программ.

**Отличия отказов ПО** при его эксплуатации от отказов технических элементов

1. Отказ ПО не приводит к разрушению или поломке программного элемента. Отказы ПО не связаны с физическим износом элемента (в частности носителя программ),
2. Отказ ПО не коррелирован с процессорным и, тем более, астрономическим временем (с процессорным временем или числом прогонов ПО программ пользователем),
3. При длительной эксплуатации ПО все его ошибки могут быть устранены и программы становятся абсолютно надёжными. Если обозначить через N(t) число не выявленных ошибок ПО в произвольный момент процессорного времени t, то формально имеет место соотношение lim N(t) = 0, справедливо при условии, что в процессе восстановления программ в них не вносятся новые ошибки.

Опыт создания и эксплуатации ПО реальному времени показывает, что **при устранении одних ошибок вносятся другие.** Поэтому, при длительной

…

Основные показатели надёжности ПО

1. **Функция ненадёжности** или **отказа**. ПО Q(t) = P (того, что отказ ПО появится до момента времени t),
2. **Функция надёжности.** ПО P(t) = P (того, что отказ ПО появится после момента времени t),
3. **Интенсивность отказов** ПО: λ(t) = dQ/dt,
4. **Средняя наработка на отказ** ПО: t = интеграл (t \* f(t) / dt) = интеграл ((P(t) \* d(t))

Программное обеспечение АСУТП состоит из …

Если программы имеют взаимные информационные связи или предназначены для получения одного результата y (вычисление одной функции), то в отношении надёжности такой программный комплекс представляет собой простую систему без избыточности и вероятность его безотказной работы равна произведению вероятностей безотказной работы каждой i-ой программы?

P(t) = П(от 1 до m) pi (t), где

m – общее число программ.

Надёжность такого ПО определяется надёжностью отказов самой «ненадёжной» программы, имеющей наибольшее значение …

Понижению интенсивности λ способствует и переход на другой более высокий ЯП. Другой путь повышения надёжности ПО связано с резервированием и введением в программную систему некоторой избыточности.

Применительно к ПО АСУТП различают **три вида резервирования**:

1. **Временное**
2. **Информационное**
3. **Программное**

**Типы данных**

**Операционные системы**

**Языки высокого уровня**

**Язык низского уровня**

**Щлюз**

**Сетевой протокол**

**Вот краткие объяснения для каждого из упомянутых терминов:**

**1. \*\*Типы данных\*\* — это классификация данных, используемых в программировании для указания компилятору или интерпретатору, как программист намерен использовать данные. Основные типы данных:**

**- \*\*Примитивные\*\*: например, `int` (целые числа), `float` (вещественные числа), `char` (символы), `boolean` (логические значения).**

**- \*\*Структурные\*\*: массивы, списки, структуры.**

**- \*\*Пользовательские\*\*: классы и структуры, определенные разработчиками.**

**2. \*\*Операционные системы (ОС)\*\* — это программное обеспечение, которое управляет аппаратными ресурсами компьютера и предоставляет интерфейсы для взаимодействия пользователя с системой. Примеры ОС:**

**- \*\*Windows\*\***

**- \*\*Linux\*\***

**- \*\*macOS\*\***

**- \*\*Android\*\***

**3. \*\*Языки высокого уровня\*\* — это языки программирования, которые предоставляют абстракции от машинного кода и удобны для написания программ. Они ближе к человеческому языку и автоматически управляют многими аспектами работы с памятью. Примеры:**

**- \*\*Python\*\***

**- \*\*Java\*\***

**- \*\*C++\*\***

**- \*\*JavaScript\*\***

**4. \*\*Язык низкого уровня\*\* — это язык программирования, который близок к машинному коду и обеспечивает высокую степень контроля над оборудованием. Примеры:**

**- \*\*Ассемблер\*\* — язык, напрямую связанный с архитектурой процессора.**

**- \*\*Машинный код\*\* — двоичные команды, которые процессор может выполнять напрямую.**

**5. \*\*Шлюз (Gateway)\*\* — это устройство или программа, которая соединяет две разные сети и управляет их взаимодействием. Шлюз может преобразовывать форматы данных, протоколы и может работать на разных уровнях модели OSI. Например, шлюз может соединять локальную сеть с интернетом.**

**6. \*\*Сетевой протокол\*\* — это набор правил, определяющих, как данные передаются по сети. Примеры сетевых протоколов:**

**- \*\*TCP/IP\*\* — основной набор протоколов для интернета.**

**- \*\*HTTP\*\* — протокол передачи гипертекста, используемый в веб-браузерах для взаимодействия с веб-серверами.**

**- \*\*FTP\*\* — протокол передачи файлов.**

**- \*\*DNS\*\* — протокол преобразования доменных имен в IP-адреса.**