

# **Методы тестирования на соответствие стандартам, обеспечивающим переносимость прикладных программ**

## **5.1 Общие требования к методам тестирования**

Методы тестирования должны удовлетворять всем критериям, описанным ниже:

- методы тестирования должны документировать спецификации методов тестирования стандарта POSIX.n, которому они соответствуют;
- методы тестирования должны проверять основные утверждения на предмет существования требуемых свойств и те условные свойства, для которых определяется степень соответствия стандарту POSIX. Если в списке тестовых утверждений содержится набор пунктов, разделенных логическим ИЛИ, то метод тестирования должен проверить каждый из этих пунктов на соответствие стандарту;
- набор программ для тестирования (НПТС) на соответствие стандартам POSIX должен быть максимально автоматизированным;
- документация НПТС должна включать всю необходимую информацию для инсталляции, конфигурирования и выполнения НПТС;
- документация методов тестирования должна включать, по мере необходимости, инструкции по выполнению процедуры тестирования на соответствие стандартам POSIX (ПТС);
- документация методов тестирования должна описывать, как обобщаются и интерпретируются результаты тестирования.

## **5.2 Уровни тестирования и уровни сложности**

5.2.1 Цель проведения тестирования на соответствие состоит в том, чтобы установить удовлетворяет ли проверяемая ПП техническим требованиям, описанным в соответствующем стандарте POSIX.

Ограничения, связанные с ресурсами компьютеров и затраченным временем на тестирование, делают невозможным проведение исчерпывающих проверок. Из-за этих ограничений, в процессе разработки методов тестирования уровень сложности проверяемого элемента определяет уровень тестирования, который удовлетворял бы требованиям, содержащимся в настоящем стандарте.

5.2.2. Уровни тестирования:

- исчерпывающий тест;
- полный тест;
- идентифицирующий тест.

5.2.2.1 Исчерпывающий тест стремится проверить поведение всех возможных состояний тестируемого элемента, включая любые перестановки. Перебор различных опций команды и перебор всех возможных комбинаций параметров в некоторых случаях требует проведения огромного числа испытаний, что делает невозможным проведение исчерпывающего теста в течение допустимого отрезка времени.

5.2.2.2 Полный тест является одним из упрощенных вариантов исчерпывающего теста, который обычно используется на практике. При выполнении полного теста стремятся проверять поведение каждого состояния элемента, но не осуществлять проверку всех перестановок. В процессе проведения полного теста набора элементов, тестирование проходит и большинство подэлементов. В этом смысле полезно рассматривать только то число тестовых утверждений, которое может быть выделено из спецификации каждого элемента прикладной программы. Если число тестируемых элементов является настолько большим, что становится невозможным проведение их полного теста, используют третий уровень тестирования.

5.2.2.3 Идентифицирующий тест обычно проверяет только отличительную особенность рассматриваемого элемента. Тестирование элемента проводится при использовании минимально возможного синтаксиса команд и проверки минимального количества функций.

5.2.3 Уровни сложности элементов тестирования:

- простые элементы;
- элементы промежуточной сложности;
- сложные элементы.

5.2.3.1 Простые элементы - элементы, полностью определенные в своих описаниях, и должны проверяться с использованием полного теста. Простые элементы обычно содержат небольшое число тестовых утверждений и не зависят от других элементов, определенных в том же стандарте POSIX.

**Пример** - Утилита `cat`, описанная в стандарте POSIX.1 или функция `close ( )`, определенная в POSIX.1.

5.2.3.2 Элементы промежуточной сложности содержат большее число тестовых утверждений, чем простые элементы, и могут зависеть от других элементов, определенных в соответствующем стандарте POSIX. Проведение для таких элементов полного теста в некоторых случаях является невозможным.

**Пример** - Утилиты `grep` и `sed`, определенные в стандарте POSIX.2.

### 5.2.3.3 Сложные элементы

К сложным элементам относятся элементы, которые могут либо поддерживать собственный язык команд (например, утилита `bash`), либо являются функциями элементов промежуточной сложности, либо работают с аппаратной частью. Для сложных элементов проведение полного тестирования возможно только для их составных частей.

*Пример* - Утилиты `sh` и `awk`, определенные в POSIX.2.

## 5.3 Классификация тестовых утверждений

5.3.1 Каждое из тестовых утверждений относится к одной из четырех категорий, представленных в виде матрицы 2x2 и изображенных в таблице 1.

**Таблица 1** - Классификация тестовых утверждений

Свойство	Утверждение	
	Основное	Расширенное
Требуемое	A	B
Условное	C	D

Строки матрицы определяют является ли свойство требуемым или условным. Столбцы матрицы определяют должно ли тестироваться данное свойство.

5.3.2 Тестовые утверждения, проверка которых необязательна, называются расширенными тестовыми утверждениями. Тестовое утверждение может быть классифицировано как расширенное, если:

- не существует известных методов тестирования данного утверждения, обладающих свойствами переносимости;
- соответствующее утверждение в стандарте POSIX недостаточно точно сформулировано, чтобы написать соответствующий тест;
- не существует известных достоверных методов тестирования данного утверждения;
- проверка тестового утверждения требует неоправданно больших трудозатрат.

Проверка расширенных тестовых утверждений осуществляется по мере возможности.

В некоторых случаях возможно проведение тестирования не целого расширенного утверждения, а лишь его части.

**Пример** - Возможно определить проявление ошибки на частных элементах, но невозможно определить все возможные ошибки, возникающие в этих элементах.

Проверка тестового утверждения, определившая ошибку, должна возвращать код результата "FAIL". Любая проверка тестового утверждения, которая по каким бы то ни было причинам не завершилась корректно, должна возвращать код результата тестирования "UNTESTED".

5.3.3 Тестовые утверждения, которые не являются расширенными, называются основными тестовыми утверждениями. Основные тестовые утверждения обязательно должны проверяться как для требуемых, так и для условных свойств.

## 5.4 Написание тестовых утверждений

### 5.4.1 Определение тестового утверждения

5.4.1.1 Тестовое утверждение формируется для каждого строгого утверждения, описанного в стандарте, которое встречается в тестируемой ПП. Соответствующий тест разрабатывается с целью определения, является ли утверждение, определенное в стандарте POSIX, истинным или ложным для ПП.

5.4.1.2 Стиль написания стандартов POSIX не позволяет осуществить прямое копирование утверждения из стандарта POSIX для последующего тестирования. Каждое тестовое утверждение должно быть независимым (автономным). Тестовое утверждение должно быть сформулировано так, чтобы код результата проверки тестового утверждения "PASS" соответствовал ситуации, когда проверяемая ПП удовлетворяет требованию стандарта POSIX.

5.4.1.3 Строгим утверждением является такое утверждение, описанное в стандарте POSIX, которое содержит или подразумевает слова "будет", "должен" или "может". В этом разделе описывается, каким образом строгие утверждения стандарта POSIX могут быть интерпретированы как тестовые утверждения.

5.4.1.4 Утверждение, описанное в стандарте POSIX, которое используется в данной ПП, и содержит слово "будет" или является утвердительным предложением с глаголом "будет", является требуемым (обязательным) для данной ПП. Собственно такие утверждения, или их сочетания с другими строгими утверждениями, соответствуют тестовым утверждениям.

5.4.1.5 Утверждение, которое применяется к условному свойству ПП и использует глаголы "должен" и "может", или является предложением,

подразумевающим глаголы "должен" и "может", интерпретируется как требуемое (обязательное) для данной ПП. Собственно такие утверждения, или их сочетания с другими строгими утверждениями, соответствуют тестовым утверждениям.

5.4.1.6 Для проверки использования ПП тестовые утверждения не пишутся.

Тестовые утверждения не пишутся также и в случае, если утверждения, содержащие глаголы "должен" и "может", являются либо предупреждением для программистов, либо носят рекомендательный характер.

#### 5.4.2 Разработка тестовых утверждений

5.4.2.1 Каждое тестовое утверждение включает в себя:

- номер;
- классификатор;
- собственно текст.

5.4.2.2 Номер тестового утверждения.

Каждому тестовому утверждению присваивается номер, который является уникальным для набора тестовых утверждений, относящихся к конкретному элементу.

Тестовые утверждения должны следовать в том же порядке, что и соответствующие утверждения стандарта POSIX. Они должны быть пронумерованы последовательно, начиная с единицы (1) для каждого элемента.

Новые или измененные тестовые утверждения должны быть пронумерованы последовательно, начиная с первого свободного номера. В конце списка номеров тестовых утверждений должна присутствовать фраза "Неиспользуемые номера тестовых утверждений".

5.4.2.3 Классификатор тестовых утверждений

Тестовое утверждение, отражающее условное свойство, классифицируется как тестовое утверждение С или D типа и пишется в виде условного предложения. В противном случае тестовое утверждение имеет тип А или В.

Тестовое утверждение, тестирование которого можно осуществить переносимым НПТС за приемлемое время, классифицируется как основное тестовое утверждение (тип А или С). В противном случае оно считается расширенным тестовым утверждением (тип В или D).

Для большинства тестовых утверждений классификаторы принимают простые значения А, В, С или D (см. таблицу 1). В некоторых случаях классификация тестовых утверждений зависит от результата, возвращаемого ПП при проведении проверки. В этих случаях классификатор тестового утверждения записывается либо в виде ("СПЕЦИФИКАТОР"?А:В), либо в виде ("СПЕЦИФИКАТОР"?С:D). В первом случае, если выражение "СПЕЦИФИКАТОР" истинно, тестовое утверждение обязательного свойства классифицируется как тип А (основное), в противном случае утверждение классифицируется как В (расширенное). Во втором случае, если выражение "СПЕЦИФИКАТОР" истинно, тестовое утверждение для условного свойства классифицируется как С (основное), в противном случае - как D (расширенное).

#### 5.4.2.4 Собственно текст тестовых утверждений и примеры:

а) в простейшем случае, тестовое утверждение является простым предложением.

**Пример 1** - 01(А) Небо голубое

б) если для того, чтобы тестовое утверждение находилось во вполне определенном состоянии, требуется дополнительная информация о среде, то используется слово "Когда".

**Пример 2** - 02(А) Когда нет облаков, небо голубое

в) если тестовое утверждение записывается для условного свойства, в него включается пункт "Если", который содержит условие, следующее из стандарта POSIX. После пункта ставится двоеточие.

**Пример 3** - 03(С) Если прикладная программа поддерживает стандарт языка Си для языково-зависимых систем: Небо голубое

г) в случае, когда для тестового утверждения, которое записывается для условного свойства, необходима дополнительная информация о среде, оно представляется в виде комбинации двух конструкций, описанных выше.

**Пример 4** - 04(С) Если прикладная программа поддерживает стандарт языка Си для языково-зависимых систем: Когда нет облаков, небо голубое

д) если тестовое утверждение пишется для требуемого (обязательного) свойства, которое в свою очередь имеет различное поведение в зависимости от того, является ли конкретное условие истинным или ложным для данной ПП, используется конструкция "Если .... Иначе ...". Классификатор данного тестового утверждения может быть либо "А", либо "В".

**Пример 5 - 05(A)** Если поддерживается поведение, связанное с `{_POSIX_JOB_CONTROL}`:

Небо голубое Иначе: Небо серое

е) спецификаторы используются в том случае, когда тестовое утверждение зависит от свойств ПП.

**Пример 6 - 06({\_POSIX\_JOB\_CONTROL}?C:D)** Если прикладная программа поддерживает стандарт языка C для языково-зависимых систем: Небо голубое.

ж) если поведение или свойство нескольких функций, определенных в одной части, подобно, может быть использована конструкция "Для", позволяющая использовать один и тот же номер тестового утверждения для целого набора функций.

**Пример 7 - 07(C)** Для `earth ()`: Если прикладная программа поддерживает стандарт языка C для языково-зависимых систем: Когда нет облаков, небо голубое.

Для `moon ()`: Если прикладная программа поддерживает стандарт языка Си для языково-зависимых систем: Когда нет облаков, небо черное.

з) если реакция ПП на утверждение зависит от двух или более ее условных свойств, пишется набор тестовых утверждений, полностью перекрывающий возможные комбинации таких условных свойств.

**Пример 8 - 08(C)** Если поддерживается общий терминальный интерфейс и поддерживается поведение, связанное с `{_POSIX_JOB_CONTROL}`: Небо голубое.

**Пример 9 - 09(C)** Если общий терминальный интерфейс и поведение, связанное с `{_POSIX_JOB_CONTROL}` не поддерживаются: Небо серое.

**Пример 10 - 10(C)** Если поддерживается общий терминальный интерфейс и не поддерживается поведение, связанное с `{_POSIX_JOB_CONTROL}`: Небо черное.

и) если тестовое утверждение содержит некую величину, значение которой либо очень большое, либо не определено в стандартах POSIX, при тестировании используется новая величина, имя которой начинается с символов "PCTS". Значение этой величины используется НПТС для проведения теста.

**Пример 11 - 11(A)** Если `{OPEN_MAX}` определено и `{OPEN_MAX} < {PCTS_OPEN_MAX}`:

Небо синее. Иначе Небо серое.

к) если тестовое утверждение требует многих испытаний для того, чтобы быть проверенным полностью, к нему прилагается таблица или список минимального набора испытаний.

**Пример 12** - 12(A) Небо имеет цвета черный, белый, сине-зеленый и голубой  
Или - 12(A) Небо изменяет свой цвет в течении суток так, как показано в таблице 2.

**Таблица 2**

Время суток	Цвет
Ночь	Черный
Середина дня	Белый
Закат	Сине-зеленый

л) два тестовых утверждения записываются в случае, если стандарт POSIX определяет набор независимых начальных условий, часть из которых не может быть установлена. Для тех условий, которые могут иметь место, пишется основное тестовое утверждение, для остальных - расширенное.

**Пример 13** - Утверждение: 13(B) Либо в полдень, либо в случае разрушения Земли астероидом, система печатает слово "Привет" разбивается на два:  
13(A) В полдень система печатает слово "Привет".  
14(B) при разрушении Земли астероидом система печатает слово "Привет".

#### 5.4.3 Общие тестовые утверждения

5.4.3.1 Общие тестовые утверждения представляют собой утверждения о поведении или о функциональных возможностях, извлеченные из стандарта POSIX, и применимые более чем к одному элементу. На основе общих тестовых утверждений строятся тестовые утверждения, применимые к каждому конкретному элементу.

5.4.3.2 Общие тестовые утверждения записываются как GA#, где # - целое число, начиная с единицы (1). Значение # является уникальным для каждого общего тестового утверждения в пределах одного стандарта POSIX. Общие утверждения не имеют классификации. Каждое тестовое утверждение, полученное на основе общего тестового утверждения, имеет на него ссылку.

**Пример** - GA1 Когда нажата специальная клавиша, система останавливается

На основе этого общего тестового утверждения для клавиши BREAK можно записать:



15(A) Когда нажата клавиша BREAK, система останавливается (см. GA1).

#### 5.4.4 Ссылочные тестовые утверждения

Ссылочные тестовые утверждения записываются как R#, где # - целое число, начиная с единицы (1). Величина # является уникальной внутри описания каждого элемента. Ссылочные тестовые утверждения не имеют классификации. Ссылочные тестовые утверждения содержат ссылки на другие тестовые утверждения.

**Пример** - R01 Когда нет туч над головой, небо голубое (см. примеры 1, 2)

#### 5.4.5 Требования тестирования

Если есть вероятность того, что тестовое утверждение может быть неоднозначно или неверно интерпретировано, его конкретизация проводится введением требований тестирования, которые записываются в виде предложения после текста тестового утверждения. Требования тестирования вводятся в том случае, если попытки придать тестовому утверждению необходимую строгость только увеличивают неясность.

**Пример** - 16(A) Когда тучи закрывают небо, небо серое.  
Требования тестирования: Проверить начальный цвет

#### 5.4.6 Неиспользованные номера тестовых утверждений

Фраза "Неиспользованные номера тестовых утверждений: # - " означает, что номер утверждения # больше не используется, так как тестовое утверждение либо было отредактировано, либо было неправильным.

#### 5.4.7 Формат тестовых утверждений

Формальные требования к формату тестовых утверждений используют следующие правила:

- а) имя переменной представляет собой одно слово, состоящее из заглавных букв и цифр;
- б) переменные определяются именем переменной, стоящим перед знаком ::= и собственно определением;
- в) определение содержит упорядоченный набор переменных, литералов и текста переменной;
- г) строки, взятые в двойные кавычки ("" ) являются литералами и должны использоваться в тестовых утверждениях без каких-либо изменений;

д) переменная может иметь более одного определения. В этом случае для разграничения определений используется символ ||;

е) текст переменной записывается обычной строкой с использованием букв как в верхнем, так и в нижнем регистре.

#### 5.4.8 Правила формата тестовых утверждений

а) для тестовых утверждений типа А и В слово "если" должно всегда предшествовать слову "иначе";

б) тестовые утверждения типа С и D должны начинаться словом "если";

в) тестовые утверждения типа С и D должны начинаться словом "иначе";

г) тестовые утверждения, полученные из общего тестового утверждения, должны включать ссылку на него в форме (см. GA# в x.y.z), которая располагается за текстом тестового утверждения;

д) ссылочное тестовое утверждение должно включать указатель в форме (См. Утверждение (я) # В .y.z), который следует за текстом тестового утверждения;

е) требования тестирования должны следовать за текстом тестового утверждения после пустой строки.

### 5.5 Выходные данные проверки тестовых утверждений

5.5.1 Получаемые после выполнения проверки тестового утверждения выходные данные должны содержать:

- имя протестированного элемента;
- номер проверенного тестового утверждения,
- является ли проверенное тестовое утверждение основным или расширенным,
- код результата тестирования.

#### 5.5.2 Коды результатов тестирования

Существуют два типа кодов результатов тестирования: окончательный и промежуточный.

5.5.2.1 К окончательным кодам результатов тестирования относятся:

PASS	- после выполнения тестирования тестовое утверждение приняло значение "ИСТИНА"
------	--

FAIL	- после выполнения тестирования утверждение приняло значение "ЛОЖЬ"
UNTESTED	- возникает в случае, когда либо тест не выполнялся, либо проверка расширенного тестового утверждения не была выполнена полностью
UNSUPPORTED	- код возникает в случае, если тестирование не может быть выполнено потому, что соответствующее условное свойство не использовано

В случае, если ни один из перечисленных выше кодов результата тестирования не подходит, могут применяться дополнительные коды результата. Дополнительные коды результата могут быть использованы только теми методами тестирования, которые описаны в соответствующей документации.

5.5.2.2 К промежуточным кодам результата тестирования относится код UNRESOLVED, возникающий в случае выполнения хотя бы одного из условий:

- а) для определение результата тестирования требуется вмешательство оператора;
- б) начальные установки, требуемые для проведения тестирования, не были выполнены в необходимой форме;
- в) программа, выполняющая тестирование, была прервана;
- г) тестирование не было выполнено потому, что тестирование предыдущего утверждения, от которого зависит текущее тестовое утверждение, имело код FAIL;
- д) тестирующая программа не была инициирована;
- е) на этапе выполнения или компиляции тестирующей программы возникли непредвиденные ошибки или предупреждения;
- ж) проверка тестового утверждения не дала окончательного результата по какой-либо другой причине.

Возникновение кода UNRESOLVED должно быть обработано с целью присвоения одного из окончательных кодов результата тестирования перед вынесением заключения в виде "Сертификата соответствия".

5.5.2.3 Допустимые коды результата тестирования для каждого типа тестовых утверждений представлены в таблице 3.

Таблица 3 - Коды результата тестирования

Свойство	Тестовое утверждение	
	основное	расширенное
Требуемое (обязательное)	PASS UNRESOLVED FAIL	PASS FAIL UNTESTED UNRESOLVED
Условное	PASS UNRESOLVED UNSUPPORTED FAIL	PASS UNSUPPORTED UNTESTED UNRESOLVED FAIL

5.5.2.4 Методы тестирования не могут добавлять какой-либо другой смысл к определениям описанных выше кодов результата тестирования. Для каждого элемента существует максимальный номер используемого тестового утверждения. Для каждого номера, меньше максимального, в случае если соответствующее тестовое утверждение не используется для данного элемента, метод тестирования должен выдавать код UNRESOLVED вместо одного из описанных выше кодов результата.

5.5.2.5 Если код результата оказался равен либо FAIL, либо UNRESOLVED, метод тестирования должен обеспечить выдачу максимально полной информации для определения причины возникновения такого результата.

5.5.2.6 Если возникло непредвиденное событие, вызвавшее невозможность определения кода результата тестирования, метод тестирования должен идентифицировать место возникновения сбоя и предоставить максимально полную информацию об этом событии.

Если после возникновения непредвиденного события тестирование не может быть завершено, то код результата тестирования должен иметь значение UNRESOLVED.

5.5.2.7 Если метод тестирования представляет собой НПТС, то НПТС должен сделать попытку продолжить проверку, отметив тест, на котором произошел сбой, как UNRESOLVED.

НПТС должен пытаться избавиться от возникновения непредвиденных событий путем выполнения максимально возможного числа тестов.

