

Property-testing all the things in SerenityOS

Martin Janiczek
@janiczek

Property-testing all* the things in SerenityOS

Martin Janiczek
@janiczek

I love PBT!

I love PBT!

Property Based Testing

```
unit_test "list reversing" {  
  input = [1,2,3]  
  reversed = reverse(input)  
  assert(reversed == [3,2,1])  
}
```

```
randomized_test "list reversing" (input: List[Int]) {  
  reversed = reverse(input)  
  twice = reverse(reversed)  
  assert(twice == input)  
}
```

Property Based Testing

Test with **many random** inputs

Property Based Testing

Test with **many random** inputs

Finds a **minimal** example of a failure

Property Based Testing

Test with **many random** inputs

Finds a **minimal** example of a failure

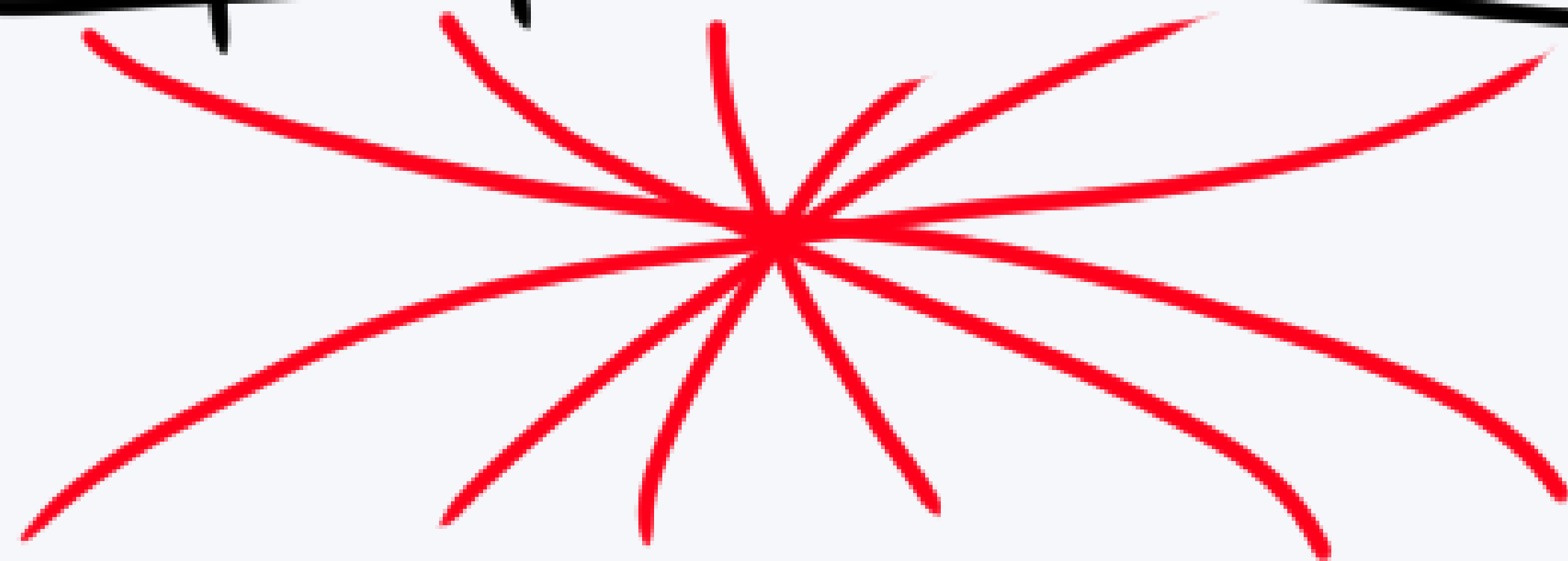
Focus on **specification**, not specific examples

```
randomized_test "list reversing" (input: List[Int]) {  
  reversed = reverse(input)  
  assert(reversed == ???)  
}
```

```
randomized_test "list reversing" (input: List[Int]) {  
  reversed = reverse(input)  
  twice = reverse(reversed)  
  assert(twice == input)  
}
```


INPUT

1	5	3	7	2	9
---	---	---	---	---	---



9	2	7	3	5	1
---	---	---	---	---	---

REVERSE (INPUT)

I love PBT!



elm-explorations / test

<> Code



Issues 78



Pull requests 8



Actions



Projects



test

Public



master



19 Branches



14 Tags

Uncommon Fuzzers

`custom` : Generator a -> Shrinker a -> Fuzzer a

Build a custom `Fuzzer a` by providing a `Generator a` and a `Shrinker a`. Generators are defined in `elm/random`. Shrinkers are defined in the `Shrink module`. It is not possible to extract the generator and shrinker from an existing fuzzer.

This function should be considered for advanced uses. It's often easier to use `map` and other functions in this module to create a fuzzer.

Here is an example for a record:

```
import Random
import Shrink

type alias Position =
  { x : Int, y : Int }

position : Fuzzer Position
position =
  Fuzz.custom
    (Random.map2 Position (Random.int -100 100) (Random.int -100 100))
    (\{ x, y } -> Shrink.map Position (Shrink.int x) |
```

Approaches

Approaches

1. QuickCheck doesn't keep constraints

Approaches

1. QuickCheck
2. Hedgehog

doesn't keep constraints

suffers when monadic bind is used

Approaches

1. QuickCheck doesn't keep constraints
2. Hedgehog suffers when monadic bind is used
3. Hypothesis ...actually pretty awesome?



2.2.0

test / CHANGELOG.md



Changes in 2.0.0

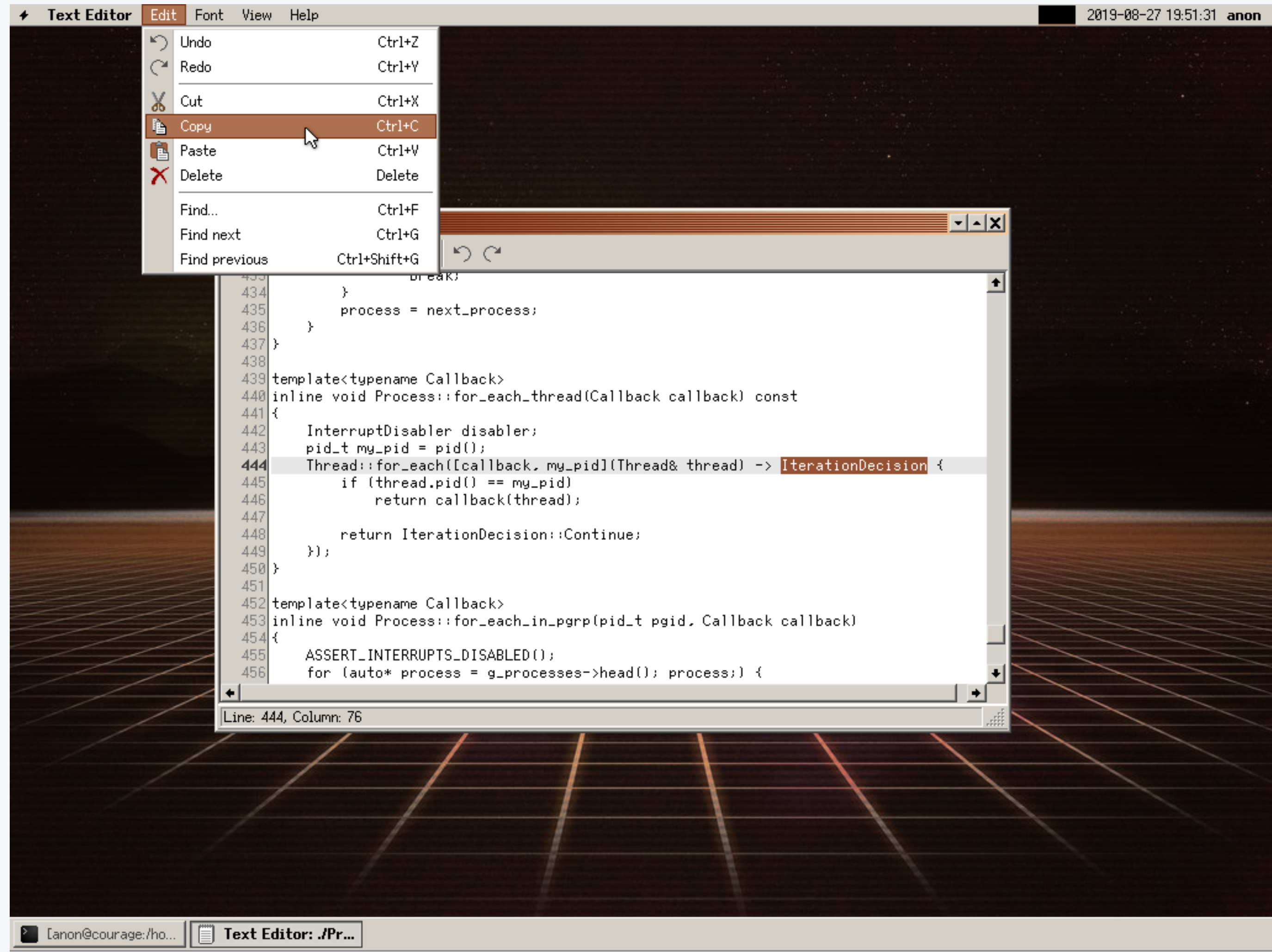
1. Fuzzing and shrinking reimplementations

Fuzzing and shrinking has been reimplemented: the rose tree approach has been replaced with the "internal shrinking" approach found in the Python test library [Hypothesis](#).

In short, shrinking is now done on the PRNG history instead of on the generated values themselves. This is hidden from the user: the `Shrink` module has now been removed.

This new approach allows us to reintroduce `Fuzz.andThen` and remove `Fuzz.custom`: in case you were forced to use `Fuzz.custom` and a `Random` generator, you'll now be able to express this logic with `Fuzz` alone.

I admire SerenityOS!



Text Editor Edit Font View Help 2019-08-27 19:51:31 anon

- Undo Ctrl+Z
- Redo Ctrl+Y
- Cut Ctrl+X
- Copy Ctrl+C
- Paste Ctrl+V
- Delete Delete
- Find.. Ctrl+F
- Find next Ctrl+G
- Find previous Ctrl+Shift+G

```
433     break;
434     }
435     process = next_process;
436 }
437 }
438
439 template<typename Callback>
440 inline void Process::for_each_thread(Callback callback) const
441 {
442     InterruptDisabler disabler;
443     pid_t my_pid = pid();
444     Thread::for_each([callback, my_pid](Thread& thread) -> IterationDecision {
445         if (thread.pid() == my_pid)
446             return callback(thread);
447
448         return IterationDecision::Continue;
449     });
450 }
451
452 template<typename Callback>
453 inline void Process::for_each_in_pgrp(pid_t pgid, Callback callback)
454 {
455     ASSERT_INTERRUPTS_DISABLED();
456     for (auto* process = g_processes->head(); process;) {
```

Line: 444, Column: 76

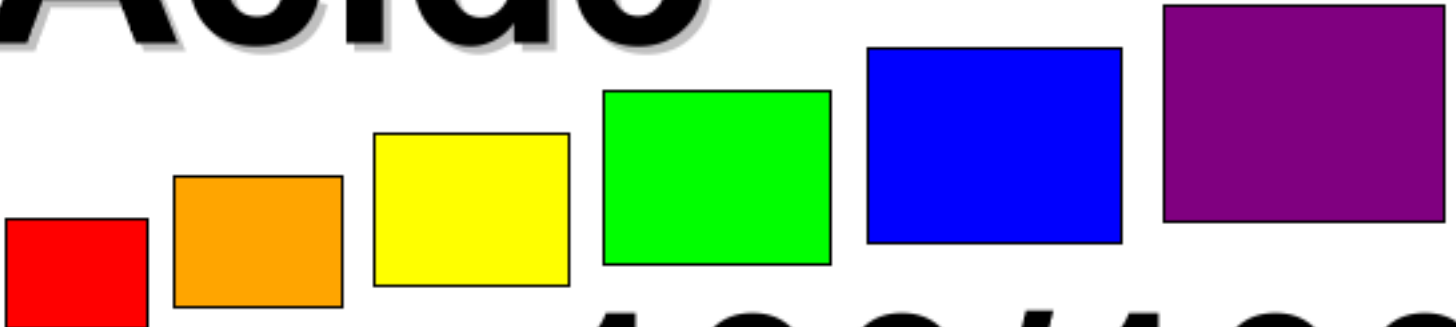
lanon@courage:/ho... Text Editor: ./Pr...

The Acid3 Test - Browser

File View Go Inspect Settings Debug Help

http://wpt.live/acid/acid3/test.html

Acid3



100/100

To pass the test, a browser must use its default settings, the animation has to be smooth, the score has to end on 100/100, and the final page has to look exactly, pixel for pixel, like [this reference rendering](#).

I love PBT!

I love PBT!

I admire SerenityOS!

I love **PBT!**

I admire **SerenityOS!**









August 24, 2023

12:53 PM **janiczek** Hey there 🙋 I wonder, would there be interest in having property-based tests for the various SerenityOS libraries and apps? I've checked TestCase and it knows about unit tests and benchmark tests... but not PBT ones. I've also seen some fuzzing via LLVM.

5:37 PM **awesomeking** hello there! I'd say we're always interested any kind of testing that can surface unknown issues :)



Janiczek / **cpp-minithesis**

<> **Code**

Issues

Pull requests

Actions



📖 **README**

cpp-minithesis



cpp-minithesis

Public

This is a port of [Minithesis](#) in C++, with the intent to try using it in SerenityOS.

Why?

What do you mean?

Why property-based testing?

It's great! Tests edge cases you didn't/couldn't think of; increases your confidence that the program works the way you think it does.

Why Minithesis instead of QuickCheck?

It uses an "internal shrinking" approach, which removes the burden of writing shrinkers from the user, and works well in face of monadic bind. This (IMHO) makes it superior to QuickCheck approach (manual/codegen'd shrinkers) and to the "integrated shrinking" lazy rose tree approach (ie. Hedgehog).



main

cpp-minithesis / main.cpp

↑ Top

Code

Blame

109 lines (96 loc) · 3.81 KB

Raw



```
3     void test_constant() {
4         run_test("constant(42) should always generate 42",
5                 Gen::constant(42),
6                 [](int num) {
7                     if (num != 42) {
8                         throw TestException("This shouldn't be possible");
9                     }
10                });
11    }
12
13    void test_constant_shrinking() {
14        run_test("constant(42) - does a failure not shrink?",
15                Gen::constant(42),
16                [](int num) { throw TestException("Should be shrunk to 42"); });
17    }
18
19    void test_unsigned_int_max_bounds() {
20        run_test("unsigned_int(10) should generate 0..10 inclusive",
21                Gen::unsigned_int(10),
22                [](unsigned int num) {
23                    if (num < 0) { throw TestException("Got something below 0: " + std::to_string(num))
24                    if (num > 10) { throw TestException("Got something above 10: " + std::to_string(num))
25                });
26    }
```



cpp-minithesis
STL

SerenityOS
AK

https://github.com/llvm/llvm-project/issues/83091

#ak



Development discussion about AK - the Agnostic Kit / Awesome Kit / Andrew Kaster / Andrew Kelley / Adorable Kittens / all kinds of files / A Keyboard / Alexander Kalenik



cpp-minithesis
STL

SerenityOS
AK

```
Test.property "bind"  
  -- generation phase:  
  (Gen.int 1 9  
    ▷ Gen.andThen (\n1 →  
      Gen.int (n1 * 10) (n1 * 100)  
      ▷ Gen.map (\n2 → (n1, n2))  
    )  
  )  
  -- testing phase:  
  (\(n1, n2) →  
    Expect.all  
      [ n1 ≥ 1 && n1 ≤ 9  
        , n2 ≥ 10 && n2 ≤ 900  
      ]  
  )
```

```

RANDOMIZED_TEST_CASE(
  bind,
  // generation phase:
  Gen::number_u64(1,9).bind([](u64 n1) {
    return Gen::number_u64(n1 * 10, n1 * 100)
      .map(=[](u64 n2){
        return Tuple<u64,u64> {n1, n2};
      });
  }),
  input
) {
  // testing phase:
  u64 n1 = input.get<0>();
  u64 n2 = input.get<1>();
  EXPECT(n1 ≥ 1 && n1 ≤ 9);
  EXPECT(n2 ≥ 10 && n2 ≤ 900);
}

```

```

Test.property "bind"
  -- generation phase:
  (Gen.int 1 9
    ▷ Gen.andThen (\n1 →
      Gen.int (n1 * 10) (n1 * 100)
        ▷ Gen.map (\n2 → (n1, n2))
    )
  )
  -- testing phase:
  (\(n1, n2) →
    Expect.all
      [ n1 ≥ 1 && n1 ≤ 9
        , n2 ≥ 10 && n2 ≤ 900
      ]
  )

```

```

RANDOMIZED_TEST_CASE(
  bind,
  // generation phase:
  Gen::number_u64(1, 9).bind([](u64 n1) {
    return Gen::number_u64(n1 * 10, n1 * 100)
      .map( [=](u64 n2){
        return Tuple<u64, u64> {n1, n2};
      });
  }),
  input
) {
  // testing phase:
  u64 n1 = input.get<0>();
  u64 n2 = input.get<1>();
  EXPECT(n1 ≥ 1 && n1 ≤ 9);
  EXPECT(n2 ≥ 10 && n2 ≤ 900);
}

```

```

RANDOMIZED_TEST_CASE(bind_like)
{
  GEN(n1, Gen::number_u64(1, 9));
  // n1 is just an int!
  EXPECT(n1 ≥ 1 && n1 ≤ 9);
  // feel free to generate again!
  GEN(n2, Gen::number_u64(n1 * 10, n1 * 100));
  EXPECT(n2 ≥ 10 && n2 ≤ 900);
}

```

```
// Values: fine!
```

```
Gen::oneOf(1, 5, 9)
```

```
// Functions: type inference sucks
```

```
Gen::oneOf([](){return Gen::number_u64(1,9);},  
           [](){return Gen::number_u64(2,3);})
```

12:15 PM CxByte @janiczek

What you want is possible, but way too complex
You'd have to zip all args and CommonType each, produce
a pack again, then splat it into Function along with the
CommonType of the return types.

Instead take the type of the first one and force everything
to be the same:

```
template <typename Fn, typename... Fns,  
typename R = decltype(declval<F>())>  
R one_of(Fn f, Fns... fns)  
{  
    Vector<Function<F>> ...;  
    ...  
}
```


LibTest: Add support for randomized tests #21191

Merged ADKaster merged 13 commits into SerenityOS/master from Jariczek/property-based-tests on Oct 27, 2023

Conversation 1/2 Commits 1/2 Checks 1/2 Files changed 1/2

Jariczek commented on Sep 22, 2023 · edited · Contributor

Add a way to run randomized tests (commonly called "property-based"): that is, tests that generate random data to run the test case with, and if they find a failure they shrink the input to a minimal failing example before reporting it to the user.

See [README.md](#) for more in-depth description.

Examples:

```
// Test\lib\Compress\test\zip.cpp
// This test didn't find anything but shows off a very common property we can test with this.
RANDOMIZED_TEST_CASE(Compress)
{
    GEN(buffer, Gen::vector(2048, []() { return (std::random_int(256)); }));
    auto compressed = MUST(Compress::ZipCompressor::compress_all(buffer));
    auto decompressed = MUST(Compress::ZipDecompressor::decompress_all(compressed));
    EXPECT(buffer == decompressed);
}

// Test\lib\2048\test\quotedPrintable.cpp
// The randomized test below found a deviation from the spec!
TEST_CASE(section_6_7_3_whitespace_regressions)
{
    // Found by the randomized test below.
    // These are the encoded tab/space at the end of the string away
    DECODE_EQUAL("1111ev", "11ev");
    DECODE_EQUAL("11ev", "11ev");

    // Doesn't throw the encoded tab/space in the middle of the string away
    DECODE_EQUAL("1111ev", "1111ev");
    DECODE_EQUAL("11ev", "11ev");
}

RANDOMIZED_TEST_CASE(section_6_7_3_whitespace)
{
    /* https://datatracker.ietf.org/doc/html/rfc2046#section-6.7
    [3] White Space! Octets with values of 9 and 32 MAY be
    represented as US-ASCII TAB (0x09) and SPACE characters,
    respectively, but MUST NOT be so represented at the end
    of an encoded line. Any TAB (0x09) or SPACE characters
    on an encoded line MUST thus be followed on that line
    by a printable character. In particular, an "*" at the
    end of an encoded line, indicating a soft line break
    (see rule #3) may follow one or more TAB (0x09) or SPACE
    characters. It follows that an octet with decimal
    value 9 or 32 appearing at the end of an encoded line
    must be represented according to Rule #3. This rule is
    necessary because some MIME (Message Transport Agents,
    programs which transport messages from one user to
    another, or perform a portion of such transfers) are
    known to pad lines of text with SPACES, and others are
    known to remove "white space" characters from the end
    of a line. Therefore, when decoding a Quoted-Printable
    body, any trailing white space on a line must be
    deleted, as it will necessarily have been added by
    intermediate transport agents.
    */

    GEN(prefix, literals_gen());
    auto prefix_sv = vector_to_string_view(prefix);

    // These are the encoded tab at the end of the string away
    Stringbuilder tab_at_end;
    tab_at_end.append(prefix_sv);
    tab_at_end.append(9);
    DECODE_EQUAL(tab_at_end.string_view(), prefix_sv);

    // These are the encoded space at the end of the string away
    Stringbuilder space_at_end;
    space_at_end.append(prefix_sv);
    space_at_end.append(32);
    DECODE_EQUAL(space_at_end.string_view(), prefix_sv);

    GEN(suffix, literals_gen());
    auto suffix_sv = vector_to_string_view(suffix);

    // Doesn't throw the encoded tab in the middle of the string away
    Stringbuilder tab_in_middle;
    tab_in_middle.append(prefix_sv);
    tab_in_middle.append(9);
    tab_in_middle.append(suffix_sv);
    Stringview tab_in_middle_sv = tab_in_middle.string_view();
    DECODE_EQUAL(tab_in_middle_sv, tab_in_middle_sv);

    // Doesn't throw the encoded space in the middle of the string away
    Stringbuilder space_in_middle;
    space_in_middle.append(prefix_sv);
    space_in_middle.append(32);
    space_in_middle.append(suffix_sv);
    Stringview space_in_middle_sv = space_in_middle.string_view();
    DECODE_EQUAL(space_in_middle_sv, space_in_middle_sv);
}
```

I'll make another PR with usages of `RANDOMIZED_TEST_CASE` that will build on this one, I appreciate this PR is already big as it is!

Note
This being my first SerenityOS contribution, I'll be glad for any suggestions (code style, C++ tricks, namespace organization!)

github-actions (bot) added the `pr-needs-review` label on Sep 22, 2023

BuggieBot commented on Sep 22, 2023

Hello!
One or more of the commit messages in this PR do not match the SerenityOS [code submission policy](#), please check the [link](#) (or click) job for more details on which commits were flagged and why.
Please do not close this PR and open another, instead modify your commit message(s) with `git commit --amend` and force push those changes to update this PR.

Jariczek force-pushed the `property-based-tests` branch from `3b74b64` to `1113b3c` 5 months ago

ADKaster commented on Sep 23, 2023

Before taking a real look at this, a few general comments from a scroll through:

- Make sure to fixup/squash any updates, or PR review comment changes.
- As written, this is one huge commit.
 - Can you split it into "atomic commits", that each change one thing, and build on top of each other to reach the final solution?
 - for example, refactorings of existing `TestMacros` or `TestCase/TestSuite` headers could be done before adding your special sauce to them, such as changing the `current_test_case_did_fail` function to `set_test_case_result[Test::Failure]`, or adding `test_result_to_string` and such in `TestMain`.
- As it is, it is one 1800 line chunk that is hard to review.
- C++ comments are preferred always. We should only have C-style comments `/* */` in the license header (for... aesthetic reasons? I actually don't know why we use that style there.)
- Full length name identifiers are preferred to shorthands in most cases. `RandomSource` -> `RandomSource`, `fn` -> `function` etc.
- The namespace of classes/functions should generally match the folder layout. In `LibTest/namespace/Tests/` in

LibTest: Add support for randomized tests #21191

Merged ADKaster merged 13 commits into SerenityOS/master from Janiczek/property-based-tests on Oct 27, 2023

Conversation 112 Commits 15 Checks 15 Files changed 15

Janiczek commented on Sep 22, 2023 · edited · Contributor

Add a way to run randomized tests (commonly called "property-based"): that is, tests that generate random data to run the test case with, and if they find a failure they shrink the input to a minimal failing example before reporting it to the user.

See [README.md](#) for more in-depth description.

Examples:

```
// Test/LibCompress/TestZip.cpp
// This test didn't find anything but shows off a very common property we can test with this.
RANDOMIZED_TEST_CASE(roundtrip)
{
    GEN(buffer, Gen::vector(2048, []() { return (std::gen::unsigned_int(256)); }));
    auto compressed = MUST(Compress::ZipCompressor::compress_all(buffer));
    auto decompressed = MUST(Compress::ZipDecompressor::decompress_all(compressed));
    EXPECT(buffer == decompressed);
}
```

```
// Test/Lib2048/TestQuotePrintable.cpp
// The randomized test below found a deviation from the spec!
TEST_CASE(section_6_7_3_whitespace_regressions)
{
    // Found by the randomized test below.
    // These are the encoded tab/space at the end of the string away
    DECODE_EQUAL("111ev", "11ev");
    DECODE_EQUAL("1ev", "1ev");
    // Doesn't throw the encoded tab/space in the middle of the string away
    DECODE_EQUAL("111ev", "111ev");
    DECODE_EQUAL("1ev", "1ev");
}
```

```
RANDOMIZED_TEST_CASE(section_6_7_3_whitespace)
{
    // https://datatracker.ietf.org/doc/html/rfc2048#section-6.7
    // | White Space| Octets with values of 9 and 32 MAY be
    // | represented as US-ASCII TAB (0x09) and SPACE characters,
    // | respectively, but MUST NOT be so represented at the end
    // | of an encoded line. Any TAB (0x09) or SPACE characters
    // | on an encoded line MUST thus be followed on that line
    // | by a printable character. In particular, an "*" at the
    // | end of an encoded line, indicating a soft line break
    // | (see rule #3) may follow one or more TAB (0x09) or SPACE
    // | characters. It follows that an octet with decimal
    // | value 9 or 32 appearing at the end of an encoded line
    // | must be represented according to Rule #3. This rule is
    // | necessary because some MIME (Message Transport Agents,
    // | programs which transport messages from one user to
    // | another, or perform a portion of such transfers) are
    // | known to pad lines of text with SPACES, and others are
    // | known to remove "white space" characters from the end
    // | of a line. Therefore, when decoding a Quoted-Printable
    // | body, any trailing white space on a line must be
    // | deleted, as it will necessarily have been added by
    // | intermediate transport agents.
    // |
    GEN(prefix, literals_gen());
    auto prefix_sv = vector_to_string_view(prefix);
    // These are the encoded tab at the end of the string away
    StringBuilder tab_at_end;
    tab_at_end.append(prefix_sv);
    tab_at_end.append(9);
    DECODE_EQUAL(tab_at_end.string_view(), prefix_sv);
    // These are the encoded space at the end of the string away
    StringBuilder space_at_end;
    space_at_end.append(prefix_sv);
    space_at_end.append(32);
    DECODE_EQUAL(space_at_end.string_view(), prefix_sv);
    GEN(suffix, literals_gen());
    auto suffix_sv = vector_to_string_view(suffix);
    // Doesn't throw the encoded tab in the middle of the string away
    StringBuilder tab_in_middle;
    tab_in_middle.append(prefix_sv);
    tab_in_middle.append(9);
    tab_in_middle.append(suffix_sv);
    StringView tab_in_middle_sv = tab_in_middle.string_view();
    DECODE_EQUAL(tab_in_middle_sv, tab_in_middle_sv);
    // Doesn't throw the encoded space in the middle of the string away
    StringBuilder space_in_middle;
    space_in_middle.append(prefix_sv);
    space_in_middle.append(32);
    space_in_middle.append(suffix_sv);
    StringView space_in_middle_sv = space_in_middle.string_view();
    DECODE_EQUAL(space_in_middle_sv, space_in_middle_sv);
}
```

I'll make another PR with usages of `RANDOMIZED_TEST_CASE` that will build on this one, I appreciate this PR is already big as it is!

Note
This being my first SerenityOS contribution, I'll be glad for any suggestions (code style, C++ tricks, namespace organization!)

github-actions (bot) added the `pr-needs-review` label on Sep 22, 2023

BuggieBot commented on Sep 22, 2023

Hello!
One or more of the commit messages in this PR do not match the SerenityOS [code submission policy](#), please check the [link](#) (click) job for more details on which commits were flagged and why.
Please do not close this PR and open another, instead modify your commit message(s) with `git commit --amend` and force push those changes to update this PR.

Janiczek force-pushed the `property-based-tests` branch from `3b78b4d` to `11a3b3c` 5 months ago

ADKaster commented on Sep 23, 2023

Before taking a real look at this, a few general comments from a scroll through:

- Make sure to fixup/squash any updates, or PR review comment changes.
- As written, this is one huge commit.
 - Can you split it into "atomic commits", that each change one thing, and build on top of each other to reach the final solution?
 - for example, refactorings of existing `TestMacros` or `TestCase/TestSuite` headers could be done before adding your special sauce to them, such as changing the `current_test_case_did_fail` function to `set_test_case_result` `Test::Failure`, or adding `test_result_to_string` and such in `TestMain`.
- As it is, it is one 1800 line chunk that is hard to review.
- C++ comments are preferred always. We should only have C-style comments `/* */` in the license header (for... aesthetic reasons? I actually don't know why we use that style there.)
- Full length name identifiers are preferred to shorthands in most cases. `RandomSource` -> `RandomSource`, `fn` -> `function` etc.
- The namespace of classes/functions should generally match the folder layout. In `LibTest/namespace/Text/()` in

LibTest/Randomized? namespace/Text/namespace/Text/ etc.

- Template parameter names should generally follow the same naming scheme as normal classes, with the obvious exception of T, U, etc (single char) (looking at you, `SET_FN`, and `FN`)
- static inline functions in headers are no bueno. It's a nice trap to duplicate the function into every TU independently.
- I wonder if your `ASSUME/REJECT` etc error cases could be modeled with `AK::EXPECT_TRY`, `EXPECT_TRY_THROW`, our common result type: `expected-like type`.
 - At least one of them for `EXPECT_TRY` could just be a `TRY_THROW` clause on the end of the template function declaration `requires (sizeof...(T) > 0.)` or similar. We are in C++20 after all.
- Some of your algorithms bear a striking resemblance to ones in AK, `seek_until` comes to mind again, `@idms180` added `any_of` in AK/AnyOf a while back, but we've been missing that drive to add generic algorithms lately...

ok maybe these aren't generic, but those things are a bit distracting from the cool feature you've added to let us validate properties of our libraries without relying on oss-fuzz :)

Janiczek force-pushed the `property-based-tests` branch 7 times, most recently from `46f4feb` to `59d1d08` 5 months ago

Janiczek mentioned this pull request on Oct 11, 2023
AK: Fix one-off error in `BitmapView::find_first` and `find_one_anywhere` #21409

timschumi self-requested a review 4 months ago

timschumi requested changes on Oct 11, 2023

timschumi left a comment

First round of feedback, mostly style-related.

Overland/Libraries/LibTest/TestResult.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Chunk.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/RandomBuf.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/RandomBuf.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/RandomBuf.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/RandomBuf.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/ShrinkCmd.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/ShrinkCmd.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Shrink.h (Outdated) Show resolved

Overland/Libraries/LibTest/Macro.h (Outdated) Show resolved

Tests/LibTest/TestGenerator.cpp (Outdated) Show resolved

Janiczek force-pushed the `property-based-tests` branch from `59d1d08` to `41672a5` 4 months ago

Janiczek requested a review from timschumi 4 months ago

Janiczek force-pushed the `property-based-tests` branch 3 times, most recently from `ac1379` to `1ffa28e` 4 months ago

ADKaster requested changes on Oct 18, 2023

ADKaster left a comment

All right, here's my nitpicks. I think most of them should be trivial, but there are a few thinker questions hidden in the noise :)

Overland/Libraries/LibTest/Macro.h (Outdated) Show resolved

Overland/Libraries/LibTest/TestSuite.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Chunk.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/RandomBuf.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/RandomBuf.h (Outdated) Show resolved

5 hidden conversations Load more...

Overland/Libraries/LibTest/Randomized/ShrinkCommand.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/ShrinkCommand.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/ShrinkCommand.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Shrink.h (Outdated) Show resolved

Overland/Libraries/LibTest/TestCase.h (Outdated) Show resolved

Janiczek force-pushed the `property-based-tests` branch from `1fa226e` to `92d8a37` 4 months ago

Janiczek requested a review from ADKaster 4 months ago

Janiczek force-pushed the `property-based-tests` branch from `92d8a37` to `2e8cc93` 4 months ago

timschumi requested changes on Oct 20, 2023

timschumi left a comment

If any of the comments here have already been resolved in advance feel free to disregard them, the first few I started writing a some days ago.

LibTest: Add support for randomized tests #2191

ADKaster merged 13 commits into SerenityOS/master from Janiczek/property-based-tests on Oct 27, 2023

Conversation 112 Commits 15 Checks 15 Files changed 15

Janiczek commented on Sep 22, 2023 · edited · Contributor

Add a way to run randomized tests (commonly called "property-based"): that is, tests that generate random data to run the test case with, and if they find a failure they shrink the input to a minimal failing example before reporting it to the user.

See [README.md](#) for more in-depth description.

Examples:

```
// Tests/LibCompress/TestZip.cpp
// This test didn't find anything but shows off a very common property we can test with this.
RANDOMIZED_TEST_CASE(example) {
    GEN(buffer, Gen::vector(2048, []() { return Gen::unsigned_int(256); }));
    auto result_compressed = MSGZCompress::GzipCompressor::compress_all(buffer);
    auto result_decompressed = MSGZCompress::GzipDecompressor::decompress_all(result_compressed);
    EXPECT(buffer == result_decompressed);
}
```

```
// Tests/LibZMQ/TestQueueOfStrings.cpp
// The randomized test below found a deviation from the spec!
```

```
TEST_CASE(section_6_7_3_whitespace_regression) {
    // Found by the randomized test below.
    // These are the encoded tab/space at the end of the string easy
    DECODE_EQUAL("1111*ev", "1*ev");
    DECODE_EQUAL("1111*ev", "1*ev");
    // Doesn't throw the encoded tab/space in the middle of the string easy
    DECODE_EQUAL("1111*ev", "1111*ev");
    DECODE_EQUAL("1111*ev", "1111*ev");
}
```

```
RANDOMIZED_TEST_CASE(section_6_7_3_whitespace) {
    // https://datatracker.ietf.org/doc/html/rfc2045#section-6.7
```

```
1) White Space) Octets with values of 9 and 32 MAY be represented as US-ASCII TAB (0x09) and SPACE characters, respectively, but MUST NOT be so represented at the end of an encoded line. Any TAB (0x09) or SPACE characters on an encoded line MUST thus be followed on that line by a printable character. In particular, an "" at the end of an encoded line, indicating a soft line break (see rule #3) may follow one or more TAB (0x09) or SPACE characters. It follows that an octet with decimal value 9 or 32 appearing at the end of an encoded line must be represented according to Rule #3. This rule is necessary because some MIME (Message Transport Agents) programs which transmit messages from one user to another, or perform a portion of such transfers) are known to pad lines of text with SPACES, and others are known to remove "white space" characters from the end of a line. Therefore, when decoding a quoted-printable body, any trailing white space on a line must be deleted, as it will necessarily have been added by intermediate transport agents.
```

```
GEN(prefix, literals_gen());
auto prefix_sv = vector_to_string_view(prefix);
```

```
// These are the encoded tab at the end of the string easy
StringBuilder tab_at_end;
tab_at_end.append(prefix_sv);
tab_at_end.append(0);
DECODE_EQUAL(tab_at_end.string_view(), prefix_sv);
```

```
// These are the encoded space at the end of the string easy
StringBuilder space_at_end;
space_at_end.append(prefix_sv);
space_at_end.append(0x20);
DECODE_EQUAL(space_at_end.string_view(), prefix_sv);
```

```
GEN(suffix, literals_gen());
auto suffix_sv = vector_to_string_view(suffix);
```

```
// Doesn't throw the encoded tab in the middle of the string easy
StringBuilder tab_in_middle;
tab_in_middle.append(prefix_sv);
tab_in_middle.append(0);
tab_in_middle.append(suffix_sv);
StringView tab_in_middle_sv = tab_in_middle.string_view();
DECODE_EQUAL(tab_in_middle_sv, tab_in_middle_sv);
```

```
// Doesn't throw the encoded space in the middle of the string easy
StringBuilder space_in_middle;
space_in_middle.append(prefix_sv);
space_in_middle.append(0x20);
space_in_middle.append(suffix_sv);
StringView space_in_middle_sv = space_in_middle.string_view();
DECODE_EQUAL(space_in_middle_sv, space_in_middle_sv);
}
```

I'll make another PR with usages of `RANDOMIZED_TEST_CASE` that will build on this one, I appreciate this PR is already big as it is!

Note
This being my first SerenityOS contribution, I'll be glad for any suggestions (code style, C++ tricks, namespace organization)

github-actions (bot) added the `pr-needs-review` label on Sep 22, 2023

BugzieBot commented on Sep 22, 2023

Hello!
One or more of the commit messages in this PR do not match the SerenityOS code submission policy, please check the [link](#) (or click) job for more details on which commits were flagged and why.
Please do not close this PR and open another, instead modify your commit message(s) with `git commit --amend` and force push those changes to update this PR.

Janiczek force-pushed the `property-based-tests` branch from `3b78b64` to `11303c5` 5 months ago

ADKaster commented on Sep 23, 2023

Before taking a real look at this, a few general comments from a scroll through:

- Make sure to fixup/squash any updates, or PR review comment changes.
- As written, this is one huge commit.
 - Can you split it into "atomic commits", that each change one thing, and build on top of each other to reach the final solution?
 - for example, refactorings of existing `TestMacros` or `TestCases/TestSuite` headers could be done before adding your special sauce to them, such as changing the `current_test_case_did_fail` function to `test_test_case_result::Test::Failure`, or adding `test_result_to_string` and such in `TestMain`.
- As it is, it is one 1800-line chunk that is hard to review.
- C++ comments are preferred always. We should only have C-style comments `/* */` in the license header (but... aesthetic reasons? I actually don't know why we use that style there.)
- Full length name identifiers are preferred in most cases. `RandomSource` -> `RandomSource`, `fn` -> `function` etc.
- The namespace of classes/functions should generally match the folder layout. In `LibTest` `namespace Test` {} in

LibTest/Randomized? namespace: Test::Randomized, etc.

- Template parameter names should generally follow the same naming scheme as normal classes, with the obvious exception of T, U, etc (single char) (looking at you, `SET_FN`, and `FN`)
- static inline functions in headers are no bueno. It's a nice trap to duplicate the function into every TU independently.
- I wonder if your `ASSUME/REJECT` etc error cases could be modeled with `REQUIRE/EXPECT`, `GENERATE/EXPECT`, our common result/void-expected-like type?
 - At least one of them, for `REQUIRE` could just be a `REQUIRE` clause on the end of the template function declaration (requires {sizeof... (T) > 0;}) or similar. We are in C++-20 after all.
- Some of your algorithms bear a striking resemblance to ones in AK, `size_of` comes to mind again, `idempotent` added `any_of` in AK/AnyOf a while back, but we've been missing that drive to add generic algorithms lately...

ok maybe these aren't generic, but those things are a bit distracting from the cool feature you've added to let us validate properties of our libraries without relying on oss-fuzz :)

Janiczek force-pushed the `property-based-tests` branch 7 times, most recently from `46f4feb` to `5941608` 5 months ago

Janiczek mentioned this pull request on Oct 11, 2023

AK: Fix one-off error in `BitmapView::find_first` and `find_one_anywhere` #21409

timschumi self-requested a review 4 months ago

timschumi requested changes on Oct 11, 2023

timschumi left a comment

First round of feedback, mostly style-related.

Overland/Libraries/LibTest/Result.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Chunk.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

timschumi left a comment

If any of the comments here have already been resolved in advance feel free to disregard them, the first few I started writing a some days ago.

Overland/Libraries/LibTest/Macros.h (Outdated) Show resolved

Overland/Libraries/LibTest/Result.h (Outdated) Show resolved

Overland/Libraries/LibTest/TestSuite.cpp (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Chunk.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

Overland/Libraries/LibTest/Randomized/Random.h (Outdated) Show resolved

timschumi on Oct 20, 2023

While we appreciate the effort, to me it looks like the comments within the actual files will be more than sufficient for explaining the structure - duplicating the information here will probably just open up opportunities for the documentation to be outdated, and they are probably closer to internals anyways.

ADKaster on Oct 20, 2023

if we really want the file, it could live in Documentation/

Janiczek on Oct 20, 2023

I felt like a summary "all in one place" would be helpful to get an overall picture, but I can remove it if you don't feel like it helped you much.

I was also contemplating another file with more hands-on examples of using this library, explaining the macros and so on. Examples of writing generators, writing the tests, etc... WDYTF?

timschumi on Oct 24, 2023

Personally, I don't mind the "Example", "Property based testing" and "Implementation" parts, but as Andrew



ADKaster approved these changes on Oct 27, 2023



ADKaster merged commit `32909d0` into `SerenityOS:master` on Oct 27, 2023

12 checks passed



ADKaster approved these changes on Oct 27, 2023



ADKaster merged commit 32909d0 into SerenityOS:master on Oct 27, 2023

12 checks passed

October 27, 2023

11:36 AM **janiczek** Whoa, I came to do another round of review fixes but MY PR IS MERGED 🐼 So cool

2:18 PM **timschumi** Now you get to make those fixups in a followup PR
timschumi :^)

timschumi at some point a 1800 line PR becomes unwieldy
2:24 PM **janiczek** Yeah I appreciate you both putting up with that one 😊

3:49 PM **Andrew Kaster** 100 comments is my limit 🤪

Examples:

```
// Tests/LibCompress/TestGzip.cpp
// This test didn't find anything but shows off a very common property we can test with this.
RANDOMIZED_TEST_CASE(roundtrip)
{
    GEN(buffer, Gen::vector(2048, []() { return (u8)Gen::unsigned_int(255); }));
    auto const compressed = MUST(Compress::GzipCompressor::compress_all(buffer));
    auto const decompressed = MUST(Compress::GzipDecompressor::decompress_all(compressed));
    EXPECT(buffer == decompressed);
}
```



Network Working Group
Request for Comments: 2045
Obsoletes: [1521](#), [1522](#), [1590](#)
Category: Standards Track

N. Freed
Innosoft
N. Borenstein
First Virtual
November 1996

Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Abstract

STD 11, [RFC 822](#), defines a message representation protocol specifying considerable detail about US-ASCII message headers, and leaves the message content, or message body, as flat US-ASCII text. This set of documents, collectively called the Multipurpose Internet Mail Extensions, or MIME, redefines the format of messages to allow for

- (1) textual message bodies in character sets other than US-ASCII,
- (2) an extensible set of different formats for non-textual message bodies,
- (3) multi-part message bodies, and
- (4) textual header information in character sets other than US-ASCII.

These documents are based on earlier work documented in [RFC 934](#), STD

Datatracker

RFC 2045

Draft Standard

Info

Contents

Prefs

Document type

RFC **Draft Standard**

November 1996

View errata

Report errata

Updated by [RFC 2184](#), [RFC 5335](#), [RFC 6532](#),
[RFC 2231](#)

Obsoletes [RFC 1590](#), [RFC 1522](#), [RFC 1521](#)

Was [draft-ietf-822ext-mime-imb](#) ([822ext WG](#))

Select version

06

RFC 2045

Compare versions

draft-ietf-822ext-mime-imb-06

RFC 2045

Side-by-side

Inline

Authors

[Ned Freed](#) ✉, [Dr. Nathaniel S. Borenstein](#) ✉

Email authors

RFC stream



I E T F

Network Working Group
Request for Comments: 2045
Obsoletes: [1521](#), [1522](#), [1590](#)
Category: Standards Track

N. Freed
Innosoft
N. Borenstein
First Virtual
November 1996

**Multipurpose Internet Mail Extensions
(MIME) Part One:
Format of Internet Message Bodies**

Datatracker
RFC 2045
Draft Standard
Info Contents Prefs
Document type
RFC Draft Standard
November 1996

AUDIENCE PARTICIPATION

and status of this protocol. Distribution of this memo is unlimited.

RFC 2045, § 6.7 (3)

- (1) textual message bodies in character sets other than US-ASCII,
- (2) an extensible set of different formats for non-textual message bodies,
- (3) multi-part message bodies, and
- (4) textual header information in character sets other than US-ASCII.

These documents are based on earlier work documented in [RFC 934](#), STD

WG)
J-06
RFC 2045
Side-by-side Inline
Authors
[Ned Freed](#) ✉, [Dr. Nathaniel S. Borenstein](#) ✉
Email authors
RFC stream
I E T F

```
GEN(prefix, literals_gen());
auto prefix_sv = vector_to_string_view(prefix);

// Throws the encoded tab at the end of the string away
StringBuilder tab_at_end;
tab_at_end.append(prefix_sv);
tab_at_end.append(9);
DECODE_EQUAL(tab_at_end.string_view(), prefix_sv);

// Throws the encoded space at the end of the string away
StringBuilder space_at_end;
space_at_end.append(prefix_sv);
space_at_end.append(32);
DECODE_EQUAL(space_at_end.string_view(), prefix_sv);
```

```
GEN(prefix, literals_gen());
auto prefix_sv = vector_to_string_view(prefix);
```

```
// Throws the encoded tab at the end of the string away
```

```
// Tests/LibIMAP/TestQuotedPrintable.cpp
// The randomized test below found a deviation from the spec!
```

```
TEST_CASE(section_6_7_3_white_space_regressions)
{
```

```
    // Found by the randomized test below.
```

```
    // Throws the encoded tab/space at the end of the string away
```

```
    DECODE_EQUAL("!\t"sv, "!"sv);
```

```
    DECODE_EQUAL("! "sv, "!"sv);
```

```
    // Doesn't throw the encoded tab/space in the middle of the string away
```

```
    DECODE_EQUAL("!\t!"sv, "!\t!"sv);
```

```
    DECODE_EQUAL("! !"sv, "! !"sv);
```

```
}
```




```
vim
1 AK:
2 TestByteBuffer.cpp
3 TestChecked.cpp
4 TestCircularBuffer.cpp
5 TestCircularDeque.cpp
6 TestCircularQueue.cpp
7 TestComplex.cpp
8 TestDeprecatedString.cpp
9 TestDisjointChunks.cpp
10 TestDistinctNumeric.cpp
11 TestDoublyLinkedList.cpp
12 TestDuration.cpp
13 TestEnumBits.cpp
14 TestFind.cpp
15 TestFixedArray.cpp
16 TestFixedPoint.cpp
17 TestFloatingPoint.cpp
18 TestFloatingPointParsing.cpp
19 TestFlyString.cpp
20 TestFormat.cpp
21 TestFuzzyMatch.cpp
22 TestGenericLexer.cpp
23 TestHashFunctions.cpp
24 TestHashMap.cpp
25 TestHashTable.cpp
26 TestHex.cpp
27 TestIPv4Address.cpp
28 TestIPv6Address.cpp
29 TestIndexSequence.cpp
30 TestInsertionSort.cpp
31 TestIntegerMath.cpp
32 TestIntrusiveList.cpp
33 TestIntrusiveRedBlackTree.cpp
34 TestJSON.cpp
35 TestLEB128.cpp
36 TestLexicalPath.cpp
37 TestMACAddress.cpp
38 TestMemory.cpp
39 TestMemoryStream.cpp
40 TestNeverDestroyed.cpp
41 TestNonnullOwnPtr.cpp
42 TestNonnullRefPtr.cpp
43 TestNumberFormat.cpp
44 TestOptional.cpp
45 TestOwnPtr.cpp
46 TestPrint.cpp
47 TestQueue.cpp
48 TestQuickSelect.cpp
49 TestQuickSort.cpp
50 TestRedBlackTree.cpp
51 TestRefPtr.cpp
52 TestSIMD.cpp
53 TestSinglyLinkedList.cpp
54 TestSourceGenerator.cpp
55 TestSourceLocation.cpp
56 TestSpan.cpp
57 TestStack.cpp
58 TestStatistics.cpp
59 TestStdLibExtras.cpp
60 TestString.cpp
61 TestStringFloatingPointConversions.cpp
62 TestStringUtils.cpp
63 TestStringView.cpp
64 TestTrie.cpp
65 TestTuple.cpp
66 TestTypeTraits.cpp
67 TestTypedTransfer.cpp
68 TestUFixedBigInt.cpp
69 TestURL.cpp
70 TestUtf16.cpp
71 TestUtf8.cpp
72 TestVariant.cpp
73 TestVector.cpp
74 TestWeakPtr.cpp
75
76 Kernel:
77 TestEFault.cpp
78 TestEmptyPrivateInodeVMObject.cpp
79 TestEmptySharedInodeVMObject.cpp
80 TestExt2FS.cpp
81 TestInvalidUIDSet.cpp
82 TestKernelAlarm.cpp
83 TestKernelFilePermissions.cpp
84 TestKernelPledge.cpp
85 TestKernelUnveil.cpp
86 TestMemoryDeviceMmap.cpp
87 TestMunMap.cpp
88 TestPosixFallocate.cpp
89 TestPrivateInodeVMObject.cpp
90 TestProcFS.cpp
91 TestProcFSWrite.cpp
92 TestSharedInodeVMObject.cpp
93 TestSigAltStack.cpp
94 TestSigHandler.cpp
95 TestSigWait.cpp
96
97 LibAudio:
98 TestFLACSpec.cpp
99 TestPlaybackStream.cpp
100
101 LibC:
102 TestAbort.cpp
103 TestAssert.cpp
104 TestCType.cpp
105 TestEnvironment.cpp
106 TestIo.cpp
107 TestLibCDirEnt.cpp
108 TestLibCExec.cpp
109 TestLibCInodeWatcher.cpp
110 TestLibCMkTemp.cpp
111 TestLibCNetdb.cpp
112 TestLibCSetjmp.cpp
113 TestLibCString.cpp
114 TestLibCTime.cpp
115 TestMalloc.cpp
116 TestMath.cpp
117
118 LibCore:
119 TestLibCoreArgsParser.cpp
120 TestLibCoreDeferredInvoke.cpp
121 TestLibCoreFilePermissionsMa
122 TestLibCoreFileWatcher.cpp
123 TestLibCoreMappedFile.cpp
124 TestLibCorePromise.cpp
125 TestLibCoreSharedSingleProdu
126 TestLibCoreStream.cpp
127
128 LibCrypto:
129 TestAES.cpp
130 TestASN1.cpp
131 TestBigInteger.cpp
132 TestChaCha20.cpp
133 TestChacha20Poly1305.cpp
134 TestChecksum.cpp
135 TestCurves.cpp
136 TestEd25519.cpp
137 TestHMAC.cpp
138 TestHash.cpp
139 TestPBKDF2.cpp
140 TestStrncpy.cpp
141 TestStrtodAccuracy.cpp
142 TestWchar.cpp
143 TestWctype.cpp
144
145 LibC++:
146 test-cpp-parser.cpp
147 test-cpp-preprocessor.cpp
148
149 LibC++:
150 TestAES.cpp
151 TestASN1.cpp
152 TestBigInteger.cpp
153 TestChaCha20.cpp
154 TestChacha20Poly1305.cpp
155 TestChecksum.cpp
156 TestCurves.cpp
157 TestEd25519.cpp
158 TestHMAC.cpp
159 TestHash.cpp
160 TestPBKDF2.cpp
161
162 LibC++:
163 TestAES.cpp
164 TestASN1.cpp
165 TestBigInteger.cpp
166 TestChaCha20.cpp
167 TestChacha20Poly1305.cpp
168 TestChecksum.cpp
169 TestCurves.cpp
170 TestEd25519.cpp
171 TestHMAC.cpp
172 TestHash.cpp
173 TestPBKDF2.cpp
174
175 LibC++:
176 TestAES.cpp
177 TestASN1.cpp
178 TestBigInteger.cpp
179 TestChaCha20.cpp
180 TestChacha20Poly1305.cpp
181 TestChecksum.cpp
182 TestCurves.cpp
183 TestEd25519.cpp
184 TestHMAC.cpp
185 TestHash.cpp
186 TestPBKDF2.cpp
187
188 LibC++:
189 TestAES.cpp
190 TestASN1.cpp
191 TestBigInteger.cpp
192 TestChaCha20.cpp
193 TestChacha20Poly1305.cpp
194 TestChecksum.cpp
195 TestCurves.cpp
196 TestEd25519.cpp
197 TestHMAC.cpp
198 TestHash.cpp
199 TestPBKDF2.cpp
200
201 LibC++:
202 TestAES.cpp
203 TestASN1.cpp
204 TestBigInteger.cpp
205 TestChaCha20.cpp
206 TestChacha20Poly1305.cpp
207 TestChecksum.cpp
208 TestCurves.cpp
209 TestEd25519.cpp
210 TestHMAC.cpp
211 TestHash.cpp
212 TestPBKDF2.cpp
213
214 LibC++:
215 TestAES.cpp
216 TestASN1.cpp
217 TestBigInteger.cpp
218 TestChaCha20.cpp
219 TestChacha20Poly1305.cpp
220 TestChecksum.cpp
221 TestCurves.cpp
222 TestEd25519.cpp
223 TestHMAC.cpp
224 TestHash.cpp
225 TestPBKDF2.cpp
226
227 LibC++:
228 TestAES.cpp
229 TestASN1.cpp
230 TestBigInteger.cpp
231 TestChaCha20.cpp
232 TestChacha20Poly1305.cpp
233 TestChecksum.cpp
234 TestCurves.cpp
235 TestEd25519.cpp
236 TestHMAC.cpp
237 TestHash.cpp
238 TestPBKDF2.cpp
239
240 LibC++:
241 TestAES.cpp
242 TestASN1.cpp
243 TestBigInteger.cpp
244 TestChaCha20.cpp
245 TestChacha20Poly1305.cpp
246 TestChecksum.cpp
247 TestCurves.cpp
248 TestEd25519.cpp
249 TestHMAC.cpp
250 TestHash.cpp
251 TestPBKDF2.cpp
252
253 LibC++:
254 TestAES.cpp
255 TestASN1.cpp
256 TestBigInteger.cpp
257 TestChaCha20.cpp
258 TestChacha20Poly1305.cpp
259 TestChecksum.cpp
260 TestCurves.cpp
261 TestEd25519.cpp
262 TestHMAC.cpp
263 TestHash.cpp
264 TestPBKDF2.cpp
265
266 LibC++:
267 TestAES.cpp
268 TestASN1.cpp
269 TestBigInteger.cpp
270 TestChaCha20.cpp
271 TestChacha20Poly1305.cpp
272 TestChecksum.cpp
273 TestCurves.cpp
274 TestEd25519.cpp
275 TestHMAC.cpp
276 TestHash.cpp
277 TestPBKDF2.cpp
278
279 LibC++:
280 TestAES.cpp
281 TestASN1.cpp
282 TestBigInteger.cpp
283 TestChaCha20.cpp
284 TestChacha20Poly1305.cpp
285 TestChecksum.cpp
286 TestCurves.cpp
287 TestEd25519.cpp
288 TestHMAC.cpp
289 TestHash.cpp
290 TestPBKDF2.cpp
291
292 LibC++:
293 TestAES.cpp
294 TestASN1.cpp
295 TestBigInteger.cpp
296 TestChaCha20.cpp
297 TestChacha20Poly1305.cpp
298 TestChecksum.cpp
299 TestCurves.cpp
300 TestEd25519.cpp
301 TestHMAC.cpp
302 TestHash.cpp
303 TestPBKDF2.cpp
304
305 LibC++:
306 TestAES.cpp
307 TestASN1.cpp
308 TestBigInteger.cpp
309 TestChaCha20.cpp
310 TestChacha20Poly1305.cpp
311 TestChecksum.cpp
312 TestCurves.cpp
313 TestEd25519.cpp
314 TestHMAC.cpp
315 TestHash.cpp
316 TestPBKDF2.cpp
317
318 LibC++:
319 TestAES.cpp
320 TestASN1.cpp
321 TestBigInteger.cpp
322 TestChaCha20.cpp
323 TestChacha20Poly1305.cpp
324 TestChecksum.cpp
325 TestCurves.cpp
326 TestEd25519.cpp
327 TestHMAC.cpp
328 TestHash.cpp
329 TestPBKDF2.cpp
330
331 LibC++:
332 TestAES.cpp
333 TestASN1.cpp
334 TestBigInteger.cpp
335 TestChaCha20.cpp
336 TestChacha20Poly1305.cpp
337 TestChecksum.cpp
338 TestCurves.cpp
339 TestEd25519.cpp
340 TestHMAC.cpp
341 TestHash.cpp
342 TestPBKDF2.cpp
343
344 LibC++:
345 TestAES.cpp
346 TestASN1.cpp
347 TestBigInteger.cpp
348 TestChaCha20.cpp
349 TestChacha20Poly1305.cpp
350 TestChecksum.cpp
351 TestCurves.cpp
352 TestEd25519.cpp
353 TestHMAC.cpp
354 TestHash.cpp
355 TestPBKDF2.cpp
356
357 LibC++:
358 TestAES.cpp
359 TestASN1.cpp
360 TestBigInteger.cpp
361 TestChaCha20.cpp
362 TestChacha20Poly1305.cpp
363 TestChecksum.cpp
364 TestCurves.cpp
365 TestEd25519.cpp
366 TestHMAC.cpp
367 TestHash.cpp
368 TestPBKDF2.cpp
369
370 LibC++:
371 TestAES.cpp
372 TestASN1.cpp
373 TestBigInteger.cpp
374 TestChaCha20.cpp
375 TestChacha20Poly1305.cpp
376 TestChecksum.cpp
377 TestCurves.cpp
378 TestEd25519.cpp
379 TestHMAC.cpp
380 TestHash.cpp
381 TestPBKDF2.cpp
382
383 LibC++:
384 TestAES.cpp
385 TestASN1.cpp
386 TestBigInteger.cpp
387 TestChaCha20.cpp
388 TestChacha20Poly1305.cpp
389 TestChecksum.cpp
390 TestCurves.cpp
391 TestEd25519.cpp
392 TestHMAC.cpp
393 TestHash.cpp
394 TestPBKDF2.cpp
395
396 LibC++:
397 TestAES.cpp
398 TestASN1.cpp
399 TestBigInteger.cpp
400 TestChaCha20.cpp
401 TestChacha20Poly1305.cpp
402 TestChecksum.cpp
403 TestCurves.cpp
404 TestEd25519.cpp
405 TestHMAC.cpp
406 TestHash.cpp
407 TestPBKDF2.cpp
408
409 LibC++:
410 TestAES.cpp
411 TestASN1.cpp
412 TestBigInteger.cpp
413 TestChaCha20.cpp
414 TestChacha20Poly1305.cpp
415 TestChecksum.cpp
416 TestCurves.cpp
417 TestEd25519.cpp
418 TestHMAC.cpp
419 TestHash.cpp
420 TestPBKDF2.cpp
421
422 LibC++:
423 TestAES.cpp
424 TestASN1.cpp
425 TestBigInteger.cpp
426 TestChaCha20.cpp
427 TestChacha20Poly1305.cpp
428 TestChecksum.cpp
429 TestCurves.cpp
430 TestEd25519.cpp
431 TestHMAC.cpp
432 TestHash.cpp
433 TestPBKDF2.cpp
434
435 LibC++:
436 TestAES.cpp
437 TestASN1.cpp
438 TestBigInteger.cpp
439 TestChaCha20.cpp
440 TestChacha20Poly1305.cpp
441 TestChecksum.cpp
442 TestCurves.cpp
443 TestEd25519.cpp
444 TestHMAC.cpp
445 TestHash.cpp
446 TestPBKDF2.cpp
447
448 LibC++:
449 TestAES.cpp
450 TestASN1.cpp
451 TestBigInteger.cpp
452 TestChaCha20.cpp
453 TestChacha20Poly1305.cpp
454 TestChecksum.cpp
455 TestCurves.cpp
456 TestEd25519.cpp
457 TestHMAC.cpp
458 TestHash.cpp
459 TestPBKDF2.cpp
460
461 LibC++:
462 TestAES.cpp
463 TestASN1.cpp
464 TestBigInteger.cpp
465 TestChaCha20.cpp
466 TestChacha20Poly1305.cpp
467 TestChecksum.cpp
468 TestCurves.cpp
469 TestEd25519.cpp
470 TestHMAC.cpp
471 TestHash.cpp
472 TestPBKDF2.cpp
473
474 LibC++:
475 TestAES.cpp
476 TestASN1.cpp
477 TestBigInteger.cpp
478 TestChaCha20.cpp
479 TestChacha20Poly1305.cpp
480 TestChecksum.cpp
481 TestCurves.cpp
482 TestEd25519.cpp
483 TestHMAC.cpp
484 TestHash.cpp
485 TestPBKDF2.cpp
486
487 LibC++:
488 TestAES.cpp
489 TestASN1.cpp
490 TestBigInteger.cpp
491 TestChaCha20.cpp
492 TestChacha20Poly1305.cpp
493 TestChecksum.cpp
494 TestCurves.cpp
495 TestEd25519.cpp
496 TestHMAC.cpp
497 TestHash.cpp
498 TestPBKDF2.cpp
499
500 LibC++:
501 TestAES.cpp
502 TestASN1.cpp
503 TestBigInteger.cpp
504 TestChaCha20.cpp
505 TestChacha20Poly1305.cpp
506 TestChecksum.cpp
507 TestCurves.cpp
508 TestEd25519.cpp
509 TestHMAC.cpp
510 TestHash.cpp
511 TestPBKDF2.cpp
512
513 LibC++:
514 TestAES.cpp
515 TestASN1.cpp
516 TestBigInteger.cpp
517 TestChaCha20.cpp
518 TestChacha20Poly1305.cpp
519 TestChecksum.cpp
520 TestCurves.cpp
521 TestEd25519.cpp
522 TestHMAC.cpp
523 TestHash.cpp
524 TestPBKDF2.cpp
525
526 LibC++:
527 TestAES.cpp
528 TestASN1.cpp
529 TestBigInteger.cpp
530 TestChaCha20.cpp
531 TestChacha20Poly1305.cpp
532 TestChecksum.cpp
533 TestCurves.cpp
534 TestEd25519.cpp
535 TestHMAC.cpp
536 TestHash.cpp
537 TestPBKDF2.cpp
538
539 LibC++:
540 TestAES.cpp
541 TestASN1.cpp
542 TestBigInteger.cpp
543 TestChaCha20.cpp
544 TestChacha20Poly1305.cpp
545 TestChecksum.cpp
546 TestCurves.cpp
547 TestEd25519.cpp
548 TestHMAC.cpp
549 TestHash.cpp
550 TestPBKDF2.cpp
551
552 LibC++:
553 TestAES.cpp
554 TestASN1.cpp
555 TestBigInteger.cpp
556 TestChaCha20.cpp
557 TestChacha20Poly1305.cpp
558 TestChecksum.cpp
559 TestCurves.cpp
560 TestEd25519.cpp
561 TestHMAC.cpp
562 TestHash.cpp
563 TestPBKDF2.cpp
564
565 LibC++:
566 TestAES.cpp
567 TestASN1.cpp
568 TestBigInteger.cpp
569 TestChaCha20.cpp
570 TestChacha20Poly1305.cpp
571 TestChecksum.cpp
572 TestCurves.cpp
573 TestEd25519.cpp
574 TestHMAC.cpp
575 TestHash.cpp
576 TestPBKDF2.cpp
577
578 LibC++:
579 TestAES.cpp
580 TestASN1.cpp
581 TestBigInteger.cpp
582 TestChaCha20.cpp
583 TestChacha20Poly1305.cpp
584 TestChecksum.cpp
585 TestCurves.cpp
586 TestEd25519.cpp
587 TestHMAC.cpp
588 TestHash.cpp
589 TestPBKDF2.cpp
590
591 LibC++:
592 TestAES.cpp
593 TestASN1.cpp
594 TestBigInteger.cpp
595 TestChaCha20.cpp
596 TestChacha20Poly1305.cpp
597 TestChecksum.cpp
598 TestCurves.cpp
599 TestEd25519.cpp
600 TestHMAC.cpp
601 TestHash.cpp
602 TestPBKDF2.cpp
603
604 LibC++:
605 TestAES.cpp
606 TestASN1.cpp
607 TestBigInteger.cpp
608 TestChaCha20.cpp
609 TestChacha20Poly1305.cpp
610 TestChecksum.cpp
611 TestCurves.cpp
612 TestEd25519.cpp
613 TestHMAC.cpp
614 TestHash.cpp
615 TestPBKDF2.cpp
616
617 LibC++:
618 TestAES.cpp
619 TestASN1.cpp
620 TestBigInteger.cpp
621 TestChaCha20.cpp
622 TestChacha20Poly1305.cpp
623 TestChecksum.cpp
624 TestCurves.cpp
625 TestEd25519.cpp
626 TestHMAC.cpp
627 TestHash.cpp
628 TestPBKDF2.cpp
629
630 LibC++:
631 TestAES.cpp
632 TestASN1.cpp
633 TestBigInteger.cpp
634 TestChaCha20.cpp
635 TestChacha20Poly1305.cpp
636 TestChecksum.cpp
637 TestCurves.cpp
638 TestEd25519.cpp
639 TestHMAC.cpp
640 TestHash.cpp
641 TestPBKDF2.cpp
642
643 LibC++:
644 TestAES.cpp
645 TestASN1.cpp
646 TestBigInteger.cpp
647 TestChaCha20.cpp
648 TestChacha20Poly1305.cpp
649 TestChecksum.cpp
650 TestCurves.cpp
651 TestEd25519.cpp
652 TestHMAC.cpp
653 TestHash.cpp
654 TestPBKDF2.cpp
655
656 LibC++:
657 TestAES.cpp
658 TestASN1.cpp
659 TestBigInteger.cpp
660 TestChaCha20.cpp
661 TestChacha20Poly1305.cpp
662 TestChecksum.cpp
663 TestCurves.cpp
664 TestEd25519.cpp
665 TestHMAC.cpp
666 TestHash.cpp
667 TestPBKDF2.cpp
668
669 LibC++:
670 TestAES.cpp
671 TestASN1.cpp
672 TestBigInteger.cpp
673 TestChaCha20.cpp
674 TestChacha20Poly1305.cpp
675 TestChecksum.cpp
676 TestCurves.cpp
677 TestEd25519.cpp
678 TestHMAC.cpp
679 TestHash.cpp
680 TestPBKDF2.cpp
681
682 LibC++:
683 TestAES.cpp
684 TestASN1.cpp
685 TestBigInteger.cpp
686 TestChaCha20.cpp
687 TestChacha20Poly1305.cpp
688 TestChecksum.cpp
689 TestCurves.cpp
690 TestEd25519.cpp
691 TestHMAC.cpp
692 TestHash.cpp
693 TestPBKDF2.cpp
694
695 LibC++:
696 TestAES.cpp
697 TestASN1.cpp
698 TestBigInteger.cpp
699 TestChaCha20.cpp
700 TestChacha20Poly1305.cpp
701 TestChecksum.cpp
702 TestCurves.cpp
703 TestEd25519.cpp
704 TestHMAC.cpp
705 TestHash.cpp
706 TestPBKDF2.cpp
707
708 LibC++:
709 TestAES.cpp
710 TestASN1.cpp
711 TestBigInteger.cpp
712 TestChaCha20.cpp
713 TestChacha20Poly1305.cpp
714 TestChecksum.cpp
715 TestCurves.cpp
716 TestEd25519.cpp
717 TestHMAC.cpp
718 TestHash.cpp
719 TestPBKDF2.cpp
720
721 LibC++:
722 TestAES.cpp
723 TestASN1.cpp
724 TestBigInteger.cpp
725 TestChaCha20.cpp
726 TestChacha20Poly1305.cpp
727 TestChecksum.cpp
728 TestCurves.cpp
729 TestEd25519.cpp
730 TestHMAC.cpp
731 TestHash.cpp
732 TestPBKDF2.cpp
733
734 LibC++:
735 TestAES.cpp
736 TestASN1.cpp
737 TestBigInteger.cpp
738 TestChaCha20.cpp
739 TestChacha20Poly1305.cpp
740 TestChecksum.cpp
741 TestCurves.cpp
742 TestEd25519.cpp
743 TestHMAC.cpp
744 TestHash.cpp
745 TestPBKDF2.cpp
746
747 LibC++:
748 TestAES.cpp
749 TestASN1.cpp
750 TestBigInteger.cpp
751 TestChaCha20.cpp
752 TestChacha20Poly1305.cpp
753 TestChecksum.cpp
754 TestCurves.cpp
755 TestEd25519.cpp
756 TestHMAC.cpp
757 TestHash.cpp
758 TestPBKDF2.cpp
759
760 LibC++:
761 TestAES.cpp
762 TestASN1.cpp
763 TestBigInteger.cpp
764 TestChaCha20.cpp
765 TestChacha20Poly1305.cpp
766 TestChecksum.cpp
767 TestCurves.cpp
768 TestEd25519.cpp
769 TestHMAC.cpp
770 TestHash.cpp
771 TestPBKDF2.cpp
772
773 LibC++:
774 TestAES.cpp
775 TestASN1.cpp
776 TestBigInteger.cpp
777 TestChaCha20.cpp
778 TestChacha20Poly1305.cpp
779 TestChecksum.cpp
780 TestCurves.cpp
781 TestEd25519.cpp
782 TestHMAC.cpp
783 TestHash.cpp
784 TestPBKDF2.cpp
785
786 LibC++:
787 TestAES.cpp
788 TestASN1.cpp
789 TestBigInteger.cpp
790 TestChaCha20.cpp
791 TestChacha20Poly1305.cpp
792 TestChecksum.cpp
793 TestCurves.cpp
794 TestEd25519.cpp
795 TestHMAC.cpp
796 TestHash.cpp
797 TestPBKDF2.cpp
798
799 LibC++:
800 TestAES.cpp
801 TestASN1.cpp
802 TestBigInteger.cpp
803 TestChaCha20.cpp
804 TestChacha20Poly1305.cpp
805 TestChecksum.cpp
806 TestCurves.cpp
807 TestEd25519.cpp
808 TestHMAC.cpp
809 TestHash.cpp
810 TestPBKDF2.cpp
811
812 LibC++:
813 TestAES.cpp
814 TestASN1.cpp
815 TestBigInteger.cpp
816 TestChaCha20.cpp
817 TestChacha20Poly1305.cpp
818 TestChecksum.cpp
819 TestCurves.cpp
820 TestEd25519.cpp
821 TestHMAC.cpp
822 TestHash.cpp
823 TestPBKDF2.cpp
824
825 LibC++:
826 TestAES.cpp
827 TestASN1.cpp
828 TestBigInteger.cpp
829 TestChaCha20.cpp
830 TestChacha20Poly1305.cpp
831 TestChecksum.cpp
832 TestCurves.cpp
833 TestEd25519.cpp
834 TestHMAC.cpp
835 TestHash.cpp
836 TestPBKDF2.cpp
837
838 LibC++:
839 TestAES.cpp
840 TestASN1.cpp
841 TestBigInteger.cpp
842 TestChaCha20.cpp
843 TestChacha20Poly1305.cpp
844 TestChecksum.cpp
845 TestCurves.cpp
846 TestEd25519.cpp
847 TestHMAC.cpp
848 TestHash.cpp
849 TestPBKDF2.cpp
850
851 LibC++:
852 TestAES.cpp
853 TestASN1.cpp
854 TestBigInteger.cpp
855 TestChaCha20.cpp
856 TestChacha20Poly1305.cpp
857 TestChecksum.cpp
858 TestCurves.cpp
859 TestEd25519.cpp
860 TestHMAC.cpp
861 TestHash.cpp
862 TestPBKDF2.cpp
863
864 LibC++:
865 TestAES.cpp
866 TestASN1.cpp
867 TestBigInteger.cpp
868 TestChaCha20.cpp
869 TestChacha20Poly1305.cpp
870 TestChecksum.cpp
871 TestCurves.cpp
872 TestEd25519.cpp
873 TestHMAC.cpp
874 TestHash.cpp
875 TestPBKDF2.cpp
876
877 LibC++:
878 TestAES.cpp
879 TestASN1.cpp
880 TestBigInteger.cpp
881 TestChaCha20.cpp
882 TestChacha20Poly1305.cpp
883 TestChecksum.cpp
884 TestCurves.cpp
885 TestEd25519.cpp
886 TestHMAC.cpp
887 TestHash.cpp
888 TestPBKDF2.cpp
889
890 LibC++:
891 TestAES.cpp
892 TestASN1.cpp
893 TestBigInteger.cpp
894 TestChaCha20.cpp
895 TestChacha20Poly1305.cpp
896 TestChecksum.cpp
897 TestCurves.cpp
898 TestEd25519.cpp
899 TestHMAC.cpp
900 TestHash.cpp
901 TestPBKDF2.cpp
902
903 LibC++:
904 TestAES.cpp
905 TestASN1.cpp
906 TestBigInteger.cpp
907 TestChaCha20.cpp
908 TestChacha20Poly1305.cpp
909 TestChecksum.cpp
910 TestCurves.cpp
911 TestEd25519.cpp
912 TestHMAC.cpp
913 TestHash.cpp
914 TestPBKDF2.cpp
915
916 LibC++:
917 TestAES.cpp
918 TestASN1.cpp
919 TestBigInteger.cpp
920 TestChaCha20.cpp
921 TestChacha20Poly1305.cpp
922 TestChecksum.cpp
923 TestCurves.cpp
924 TestEd25519.cpp
925 TestHMAC.cpp
926 TestHash.cpp
927 TestPBKDF2.cpp
928
929 LibC++:
930 TestAES.cpp
931 TestASN1.cpp
932 TestBigInteger.cpp
933 TestChaCha20.cpp
934 TestChacha20Poly1305.cpp
935 TestChecksum.cpp
936 TestCurves.cpp
937 TestEd25519.cpp
938 TestHMAC.cpp
939 TestHash.cpp
940 TestPBKDF2.cpp
941
942 LibC++:
943 TestAES.cpp
944 TestASN1.cpp
945 TestBigInteger.cpp
946 TestChaCha20.cpp
947 TestChacha20Poly1305.cpp
948 TestChecksum.cpp
949 TestCurves.cpp
950 TestEd25519.cpp
951 TestHMAC.cpp
952 TestHash.cpp
953 TestPBKDF2.cpp
954
955 LibC++:
956 TestAES.cpp
957 TestASN1.cpp
958 TestBigInteger.cpp
959 TestChaCha20.cpp
960 TestChacha20Poly1305.cpp
961 TestChecksum.cpp
962 TestCurves.cpp
963 TestEd25519.cpp
964 TestHMAC.cpp
965 TestHash.cpp
966 TestPBKDF2.cpp
967
968 LibC++:
969 TestAES.cpp
970 TestASN1.cpp
971 TestBigInteger.cpp
972 TestChaCha20.cpp
973 TestChacha20Poly1305.cpp
974 TestChecksum.cpp
975 TestCurves.cpp
976 TestEd25519.cpp
977 TestHMAC.cpp
978 TestHash.cpp
979 TestPBKDF2.cpp
980
981 LibC++:
982 TestAES.cpp
983 TestASN1.cpp
984 TestBigInteger.cpp
985 TestChaCha20.cpp
986 TestChacha20Poly1305.cpp
987 TestChecksum.cpp
988 TestCurves.cpp
989 TestEd25519.cpp
990 TestHMAC.cpp
991 TestHash.cpp
992 TestPBKDF2.cpp
993
994 LibC++:
995 TestAES.cpp
996 TestASN1.cpp
997 TestBigInteger.cpp
998 TestChaCha20.cpp
999 TestChacha20Poly1305.cpp
1000 TestChecksum.cpp
1001 TestCurves.cpp
1002 TestEd25519.cpp
1003 TestHMAC.cpp
1004 TestHash.cpp
1005 TestPBKDF2.cpp
1006
1007 LibC++:
1008 TestAES.cpp
1009 TestASN1.cpp
1010 TestBigInteger.cpp
1011 TestChaCha20.cpp
1012 TestChacha20Poly1305.cpp
1013 TestChecksum.cpp
1014 TestCurves.cpp
1015 TestEd25519.cpp
1016 TestHMAC.cpp
1017 TestHash.cpp
1018 TestPBKDF2.cpp
1019
1020 LibC++:
1021 TestAES.cpp
1022 TestASN1.cpp
1023 TestBigInteger.cpp
1024 TestChaCha20.cpp
1025 TestChacha20Poly1305.cpp
1026 TestChecksum.cpp
1027 TestCurves.cpp
1028 TestEd25519.cpp
1029 TestHMAC.cpp
1030 TestHash.cpp
1031 TestPBKDF2.cpp
1032
1033 LibC++:
1034 TestAES.cpp
1035 TestASN1.cpp
1036 TestBigInteger.cpp
1037 TestChaCha20.cpp
1038 TestChacha20Poly1305.cpp
1039 TestChecksum.cpp
1040 TestCurves.cpp
1041 TestEd25519.cpp
1042 TestHMAC.cpp
1043 TestHash.cpp
1044 TestPBKDF2.cpp
1045
1046 LibC++:
1047 TestAES.cpp
1048 TestASN1.cpp
1049 TestBigInteger.cpp
1050 TestChaCha20.cpp
1051 TestChacha20Poly1305.cpp
1052 TestChecksum.cpp
1053 TestCurves.cpp
1054 TestEd25519.cpp
1055 TestHMAC.cpp
1056 TestHash.cpp
1057 TestPBKDF2.cpp
1058
1059 LibC++:
1060 TestAES.cpp
1061 TestASN1.cpp
1062 TestBigInteger.cpp
1063 TestChaCha20.cpp
1064 TestChacha20Poly1305.cpp
1065 TestChecksum.cpp
1066 TestCurves.cpp
1067 TestEd25519.cpp
1068 TestHMAC.cpp
1069 TestHash.cpp
1070 TestPBKDF2.cpp
1071
1072 LibC++:
1073 TestAES.cpp
1074 TestASN1.cpp
1075 TestBigInteger.cpp
1076 TestChaCha20.cpp
1077 TestChacha20Poly1305.cpp
1078 TestChecksum.cpp
1079 TestCurves.cpp
1080 TestEd25519.cpp
1081 TestHMAC.cpp
1082 TestHash.cpp
1083 TestPBKDF2.cpp
1084
1085 LibC++:
1086 TestAES.cpp
1087 TestASN1.cpp
1088 TestBigInteger.cpp
1089 TestChaCha20.cpp
1090 TestChacha20Poly1305.cpp
1091 TestChecksum.cpp
1092 TestCurves.cpp
1093 TestEd25519.cpp
1094 TestHMAC.cpp
1095 TestHash.cpp
1096 TestPBKDF2.cpp
1097
1098 LibC++:
1099 TestAES.cpp
1100 TestASN1.cpp
1101 TestBigInteger.cpp
1102 TestChaCha20.cpp
1103 TestChacha20Poly1305.cpp
1104 TestChecksum.cpp
1105 TestCurves.cpp
1106 TestEd25519.cpp
1107 TestHMAC.cpp
1108 TestHash.cpp
1109 TestPBKDF2.cpp
1110
1111 LibC++:
1112 TestAES.cpp
1113 TestASN1.cpp
1114 TestBigInteger.cpp
1115 TestChaCha20.cpp
1116 TestChacha20Poly1305.cpp
1117 TestChecksum.cpp
1118 TestCurves.cpp
1119 TestEd25519.cpp
1120 TestHMAC.cpp
1121 TestHash.cpp
1122 TestPBKDF2.cpp
1123
1124 LibC++:
1125 TestAES.cpp
1126 TestASN1.cpp
1127 TestBigInteger.cpp
1128 TestChaCha20.cpp
1129 TestChacha20Poly1305.cpp
1130 TestChecksum.cpp
1131 TestCurves.cpp
1132 TestEd25519.cpp
1133 TestHMAC.cpp
1134 TestHash.cpp
1135 TestPBKDF2.cpp
1136
1137 LibC++:
1138 TestAES.cpp
1139 TestASN1.cpp
1140 TestBigInteger.cpp
1141 TestChaCha20.cpp
1142 TestChacha20Poly1305.cpp
1143 TestChecksum.cpp
1144 TestCurves.cpp
1145 TestEd25519.cpp
1146 TestHMAC.cpp
1147 TestHash.cpp
1148 TestPBKDF2.cpp
1149
1150 LibC++:
1151 TestAES.cpp
1152 TestASN1.cpp
1153 TestBigInteger.cpp
1154 TestChaCha20.cpp
1155 TestChacha20Poly1305.cpp
1156 TestChecksum.cpp
1157 TestCurves.cpp
1158 TestEd25519.cpp
1159 TestHMAC.cpp
1160 TestHash.cpp
1161 TestPBKDF2.cpp
1162
1163 LibC++:
1164 TestAES.cpp
1165 TestASN1.cpp
1166 TestBigInteger.cpp
1167 TestChaCha20.cpp
1168 TestChacha20Poly1305.cpp
1169 TestChecksum.cpp
1170 TestCurves.cpp
1171 TestEd25519.cpp
1172 TestHMAC.cpp
1173 TestHash.cpp
1174 TestPBKDF2.cpp
1175
1176 LibC++:
1177 TestAES.cpp
1178 TestASN1.cpp
1179 TestBigInteger.cpp
1180 TestChaCha20.cpp
1181 TestChacha20Poly1305.cpp
1182 TestChecksum.cpp
1183 TestCurves.cpp
1184 TestEd25519.cpp
1185 TestHMAC.cpp
1186 TestHash.cpp
1187 TestPBKDF2.cpp
1188
1189 LibC++:
1190 TestAES.cpp
1191 TestASN1.cpp
1192 TestBigInteger.cpp
1193 TestChaCha20.cpp
1194 TestChacha20Poly1305.cpp
1195 TestChecksum.cpp
1196 TestCurves.cpp
1197 TestEd25519.cpp
1198 TestHMAC.cpp
1199 TestHash.cpp
1200 TestPBKDF2.cpp
1201
1202 LibC++:
1203 TestAES.cpp
1204 TestASN1.cpp
1205 TestBigInteger.cpp
1206 TestChaCha20.cpp
1207 TestChacha20Poly1305.cpp
1208 TestChecksum.cpp
1209 TestCurves.cpp
1210 TestEd25519.cpp
1211 TestHMAC.cpp
1212 TestHash.cpp
1213 TestPBKDF2.cpp
1214
1215 LibC++:
1216 TestAES.cpp
1217 TestASN1.cpp
1218 TestBigInteger.cpp
1219 TestChaCha20.cpp
1220 TestChacha20Poly1305.cpp
1221 TestChecksum.cpp
1222 TestCurves.cpp
1223 TestEd25519.cpp
1224 TestHMAC.cpp
1225 TestHash.cpp
1226 TestPBKDF2.cpp
1227
1228 LibC++:
1229 TestAES.cpp
1230 TestASN1.cpp
1231 TestBigInteger.cpp
1232 TestChaCha20.cpp
1233 TestChacha20Poly1305.cpp
1234 TestChecksum.cpp
1235 TestCurves.cpp
1236 TestEd25519.cpp
1237 TestHMAC.cpp
1238 TestHash.cpp
1239 TestPBKDF2.cpp
1240
1241 LibC++:
1242 TestAES.cpp
1243 TestASN1.cpp
1244 TestBigInteger.cpp
1245 TestChaCha20.cpp
1246 TestChacha20Poly1305.cpp
1247 TestChecksum.cpp
1248 TestCurves.cpp
1249 TestEd25519.cpp
1250 TestHMAC.cpp
1251 TestHash.cpp
1252 TestPBKDF2.cpp
1253
1254 LibC++:
1255 TestAES.cpp
1256 TestASN1.cpp
1257 TestBigInteger.cpp
1258 TestChaCha20.cpp
1259 TestChacha20Poly1305.cpp
1260 TestChecksum.cpp
1261 TestCurves.cpp
1262 TestEd25519.cpp
1263 TestHMAC.cpp
1264 TestHash.cpp
1265 TestPBKDF2.cpp
1266
1267 LibC++:
1268 TestAES.cpp
1269 TestASN1.cpp
1270 TestBigInteger.cpp
1271 TestChaCha20.cpp
1272 TestChacha20Poly1305.cpp
1273 TestChecksum.cpp
1274 TestCurves.cpp
1275 TestEd25519.cpp
1276 TestHMAC.cpp
1277 TestHash.cpp
1278 TestPBKDF2.cpp
1279
1280 LibC++:
1281 TestAES.cpp
1282 TestASN1.cpp
1283 TestBigInteger.cpp
1284 TestChaCha20.cpp
1285 TestChacha20Poly1305.cpp
1286 TestChecksum.cpp
1287 TestCurves.cpp
1288 TestEd25519.cpp
1289 TestHMAC.cpp
1290 TestHash.cpp
1291 TestPBKDF2.cpp
1292
1293 LibC++:
1294 TestAES.cpp
1295 TestASN1.cpp
1296 TestBigInteger.cpp
1297 TestChaCha20.cpp
1298 TestChacha20Poly1305.cpp
1299 TestChecksum.cpp
1300 TestCurves.cpp
1301 TestEd25519.cpp
1302 TestHMAC.cpp
1303 TestHash.cpp
1304 TestPBKDF2.cpp
1305
1306 LibC++:
1307 TestAES.cpp
1308 TestASN1.cpp
1309 TestBigInteger.cpp
1310 TestChaCha20.cpp
1311 TestChacha20Poly1305.cpp
1312 TestChecksum.cpp
1313 TestCurves.cpp
1314 TestEd25519.cpp
1315 TestHMAC.cpp
1316 TestHash.cpp
1317 TestPBKDF2.cpp
1318
1319 LibC++:
1320 TestAES.cpp
1321 TestASN1.cpp
1322 TestBigInteger.cpp
1323 TestChaCha20.cpp
1324 TestChacha20Poly1305.cpp
1325 TestChecksum.cpp
1326 TestCurves.cpp
1327 TestEd25519.cpp
1328 TestHMAC.cpp
1329 TestHash.cpp
1330 TestPBKDF2.cpp
1331
1332 LibC++:
1333 TestAES.cpp
1334 TestASN1.cpp
1335 TestBigInteger.cpp
1336 TestChaCha20.cpp
1337 TestChacha20Poly1305.cpp
1338 TestChecksum.cpp
1339 TestCurves.cpp
1340 TestEd25519.cpp
1341 TestHMAC.cpp
1342 TestHash.cpp
1343 TestPBKDF2.cpp
1344
1345 LibC++:
1346 TestAES.cpp
1347 TestASN1.cpp
1348 TestBigInteger.cpp
1349 TestChaCha20.cpp
1350 TestChacha20Poly1305.cpp
1351 TestChecksum.cpp
1352 TestCurves.cpp
1353 TestEd25519.cpp
1354 TestHMAC.cpp
1355 TestHash.cpp
1356 TestPBKDF2.cpp
1357
1358 LibC++:
1359 TestAES.cpp
1360 TestASN1.cpp
1361 TestBigInteger.cpp
1362 TestChaCha20.cpp
1363 TestChacha20Poly1305.cpp
1364 TestChecksum.cpp
1365 TestCurves.cpp
1366 TestEd25519.cpp
1367 TestHMAC.cpp
1368 TestHash.cpp
1369 TestPBKDF2.cpp
1370
1371 LibC++:
1372 TestAES.cpp
1373 TestASN1.cpp
1374 TestBigInteger.cpp
1375 TestChaCha20.cpp
1376 TestChacha20Poly1305.cpp
1377 TestChecksum.cpp
1378 TestCurves.cpp
1379 TestEd25519.cpp
1380 TestHMAC.cpp
1381 TestHash.cpp
1382 TestPBKDF2.cpp
1383
1384 LibC++:
1385 TestAES.cpp
1386 TestASN1.cpp
1387 TestBigInteger.cpp
1388 TestChaCha20.cpp
1389 TestChacha20Poly1305.cpp
1390 TestChecksum.cpp
1391 TestCurves.cpp
1392 TestEd25519.cpp
1393 TestHMAC.cpp
1394 TestHash.cpp
1395 TestPBKDF2.cpp
1396
1397 LibC++:
1398 TestAES.cpp
1399 TestASN1.cpp
1400 TestBigInteger.cpp
1401 TestChaCha20.cpp
1402 TestChacha20Poly1305.cpp
1403 TestChecksum.cpp
1404 TestCurves.cpp
1405 TestEd25519.cpp
1
```

Bitmap

Bitmap

```
18
19 TEST_CASE(find_first_set)
20 {
21     auto bitmap = MUST(Bitmap::create(128, false));
22     bitmap.set(69, true);
23     EXPECT_EQ(bitmap.find_first_set().value(), 69u);
24 }
25
```


Bitmap

```
18
19 TEST_CASE(find_first_set)
20 {
21     auto bitmap = MUST(Bitmap::create(128, false));
22     bitmap.set(69, true);
23     EXPECT_EQ(bitmap.find_first_set().value(), 69u);
24 }
25
```

```
{
    auto bitmap = MUST(Bitmap::create(168, false));
    bitmap.set(34, true);
    bitmap.set(97, true);
}
```

Bitmap

```
18
19 TEST_CASE(find_first_set)
20 {
21     auto bitmap = MUST(Bitmap::create(128, false));
22     bitmap.set(69, true);
23     EXPECT_EQ(bitmap.find_first_set().value(), 69u);
24 }
25
```

```
{
    auto bitmap = MUST(Bitmap::create(288, false));
    bitmap.set_range(48, 32, true);
    bitmap.set_range(94, 39, true);
    bitmap.set_range(190, 71, true);
    bitmap.set_range(190 + 71 - 7, 21, false); // slightly overlapping clear
```

```
{
    auto bitmap = MUST(Bitmap::create(168, false));
    bitmap.set(34, true);
    bitmap.set(97, true);
```

Bitmap

```
18
19 TEST_CASE(find_first_set)
20 {
21     auto bitmap = MUST(Bitmap::create(128, false));
22     bitmap.set(69, true);
23     EXPECT_EQ(bitmap.find_first_set().value(), 69u);
24 }
25
```

```
{
    auto bitmap = MUST(Bitmap::create(288, false));
    bitmap.set_range(48, 32, true);
    bitmap.set_range(94, 39, true);
    bitmap.set_range(190, 71, true);
    bitmap.set_range(190 + 71 - 7, 21, false); // slightly overlapping clear
```

```
{
    auto bitmap = MUST(Bitmap::create(168, false));
    bitmap.set(34, true);
    bitmap.set(97, true);
```

```
{
    auto bitmap = MUST(Bitmap::create(128 + 24, false));
    bitmap.set(34, true);
    bitmap.set(126, true);
```

```
363 RANDOMIZED_TEST_CASE(find_first)
364 {
365     GEN(init, Gen::boolean());
366     GEN(size, Gen::number_u64(1, 64));
367
368     GEN(new_value, Gen::boolean());
369     GEN(i, Gen::number_u64(size - 1));
370
371     auto bitmap = MUST(Bitmap::create(size, init));
372     bitmap.set(i, new_value);
373
374     Optional<size_t> result = new_value
375         ? bitmap.find_first_set()
376         : bitmap.find_first_unset();
377
378     auto expected_found_index = init == new_value ? 0 : i;
379     EXPECT_EQ(result.value(), expected_found_index);
380 }
```



```
363 RANDOMIZED_TEST_CASE(find_first)
364 {
365     GEN(init, Gen::boolean());
366     GEN(size, Gen::number_u64(1, 64));
367
368     GEN(new_value, Gen::boolean());
369     GEN(i, Gen::number_u64(size - 1));
370
371     auto bitmap = MUST(Bitmap::create(size, init));
372     bitmap.set(i, new_value);
373     Running test 'find_first'.
374     init = false
375     size = 1
376     new_value = false
377     i = 0
378     FAIL: /Users/martin/Localhost/cloned/serenity/Tests/AK/TestBitmap.cpp:370: EXPECT(result.has_value()) failed;
379     Failed test 'find_first' in 0ms
380 }
```

```
[[nodiscard]] size_t size_in_bytes() const {  
    return ceil_div(m_size, static_cast<size_t>(8));  
}
```



```
[[nodiscard]] size_t size_in_bytes() const {  
    return ceil_div(m_size, static_cast<size_t>(8));  
}
```

```
template<bool VALUE>  
Optional<size_t> find_first() const  
{  
    size_t byte_count = m_size / 8;  
    size_t i = 0;
```



```
@@ -171,7 +171,7 @@ class BitmapView {
```

171 171

```
template<bool VALUE>
```

172 172

```
Optional<size_t> find_first() const
```

173 173

```
{
```

174

-

```
size_t byte_count = m_size / 8;
```

174

+

```
size_t byte_count = size_in_bytes();
```

175

175

```
size_t i = 0;
```

176

176

177

177

```
u8 byte = VALUE ? 0x00 : 0xff;
```



AK: Fix one-off error in BitmapView::find_first and find_one_anywhere #21409

Edit

<> Code ▾

Merged timschumi merged 1 commit into SerenityOS:master from Janiczek:fix-bitmap on Oct 11, 2023

Conversation 0

Commits 1

Checks 15

Files changed 2

+21 -3



Janiczek commented on Oct 11, 2023 · edited ▾

Contributor



The mentioned functions used `m_size / 8` instead of `size_in_bytes()` (division with ceiling rounding mode), which resulted in an off-by-one error such that the functions didn't search in the last not-fully-8-bits byte.

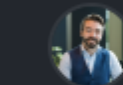
Using `size_in_bytes()` instead of `m_size / 8` fixes this.

Note

This was found using `RANDOMIZED_TEST_CASE` ([PR ready for review](#)). I'd appreciate reviews there as well, so that I could commit the [randomized tests](#) that found the issue as well!



Reviewers



gmta



timschumi



Assignees

No one assigned

Labels

None yet

Projects


```
38 TEST_CASE(Complex)
39 {
40     auto a = Complex<float> { 1.f, 1.f };
41     auto b = complex_real_unit<double> + Complex<double> { 0, 1 } * 1;
42     EXPECT_APPROXIMATE(a.real(), b.real());
43     EXPECT_APPROXIMATE(a.imag(), b.imag());
44
45 #ifdef AKCOMPLEX_CAN_USE_MATH_H
46     EXPECT_APPROXIMATE((complex_imag_unit<float> - complex_imag_unit<float>).magnitude(), 0);
47     EXPECT_APPROXIMATE((complex_imag_unit<float> + complex_real_unit<float>).magnitude(), sqrt(2));
48
49     auto c = Complex<double> { 0., 1. };
50     auto d = Complex<double>::from_polar(1., M_PI / 2.);
51     EXPECT_APPROXIMATE(c.real(), d.real());
52     EXPECT_APPROXIMATE(c.imag(), d.imag());
53
54     c = Complex<double> { -1., 1. };
55     d = Complex<double>::from_polar(sqrt(2.), 3. * M_PI / 4.);
56     EXPECT_APPROXIMATE(c.real(), d.real());
57     EXPECT_APPROXIMATE(c.imag(), d.imag());
58     EXPECT_APPROXIMATE(d.phase(), 3. * M_PI / 4.);
59     EXPECT_APPROXIMATE(c.magnitude(), d.magnitude());
60     EXPECT_APPROXIMATE(c.magnitude(), sqrt(2.));
61 #endif
62     EXPECT_EQ((complex_imag_unit<double> * complex_imag_unit<double>).real(), -1.);
63     EXPECT_EQ((complex_imag_unit<double> / complex_imag_unit<double>).real(), 1.);
64
65     EXPECT_EQ(Complex(1., 10.) == (Complex<double>(1., 0.) + Complex(0., 10.)), true);
66     EXPECT_EQ(Complex(1., 10.) != (Complex<double>(1., 1.) + Complex(0., 10.)), true);
67 #ifdef AKCOMPLEX_CAN_USE_MATH_H
68     EXPECT_EQ(approx_eq(Complex<int>(1), Complex<float>(1.0000004f)), true);
69     EXPECT_APPROXIMATE(cexp(Complex<double>(0., 1.) * M_PI).real(), -1.);
70 #endif
71 }
72
```



```
In file included from /Users/martin/Localhost/cloned/serenity/Tests/AK/TestComplex.cpp:9:
/Users/martin/Localhost/cloned/serenity/Meta/Lagom/../../AK/Complex.h:86:20: error: member reference base type 'const int' is not a structure or union
   86 |         m_real += x.real();
      |                   ~^~~~~~
/Users/martin/Localhost/cloned/serenity/Tests/AK/TestComplex.cpp:77:11: note: in instantiation of function template specialization 'AK::Complex<double>::operator+=<int>' requested here
   77 |         c += 1;
      |         ^
In file included from /Users/martin/Localhost/cloned/serenity/Tests/AK/TestComplex.cpp:9:
/Users/martin/Localhost/cloned/serenity/Meta/Lagom/../../AK/Complex.h:101:20: error: member reference base type 'const int' is not a structure or union
  101 |         m_real -= x.real();
      |                   ~^~~~~~
/Users/martin/Localhost/cloned/serenity/Tests/AK/TestComplex.cpp:82:11: note: in instantiation of function template specialization 'AK::Complex<double>::operator-=<int>' requested here
   82 |         c -= 1;
      |         ^
In file included from /Users/martin/Localhost/cloned/serenity/Tests/AK/TestComplex.cpp:9:
/Users/martin/Localhost/cloned/serenity/Meta/Lagom/../../AK/Complex.h:86:20: error: member reference base type 'const double' is not a structure or union
   86 |         m_real += x.real();
      |                   ~^~~~~~
/Users/martin/Localhost/cloned/serenity/Meta/Lagom/../../AK/Complex.h:152:11: note: in instantiation of function template specialization 'AK::Complex<double>::operator+=<double>' requested here
  152 |         x += a;
      |         ^
/Users/martin/Localhost/cloned/serenity/Tests/AK/TestComplex.cpp:166:18: note: in instantiation of function template specialization 'AK::Complex<double>::operator+<double>' requested here
  166 |         auto c2 = c1 + r2;
      |                   ^
In file included from /Users/martin/Localhost/cloned/serenity/Tests/AK/TestComplex.cpp:9:
/Users/martin/Localhost/cloned/serenity/Meta/Lagom/../../AK/Complex.h:101:20: error: member reference base type 'const double' is not a structure or union
  101 |         m_real -= x.real();
      |                   ~^~~~~~
/Users/martin/Localhost/cloned/serenity/Meta/Lagom/../../AK/Complex.h:168:11: note: in instantiation of function template specialization
```



```
In file included from /Users/martin/Localhost/cloned/serenity/Tests/AK/TestComplex.cpp:9:
/Users/martin/Localhost/cloned/serenity/Meta/Lagom/../../AK/Complex.h:86:20: error: member reference base type 'const int' is not a structure or union
```

```
86 |         m_real += x.real();
    |                   ~^~~~~~
```

```
/Users/martin/Localhost/cloned/serenity/Tests/AK/TestComplex.cpp:77:20: error: member reference base type 'const int' is not a structure or union
```

```
77 |         c += 1;
    |         ^
```

```
In file included from /Users/martin/Localhost/cloned/serenity/Tests/AK/TestComplex.cpp:9:
/Users/martin/Localhost/cloned/serenity/Meta/Lagom/../../AK/Complex.h:101:20: error: member reference base type 'const int' is not a structure or union
```

```
101 |        m_real -= x.real();
    |                   ~^~~~~~
```

```
/Users/martin/Localhost/cloned/serenity/Tests/AK/TestComplex.cpp:82:20: error: member reference base type 'const int' is not a structure or union
```

```
82 |        c -= 1;
    |        ^
```

```
In file included from /Users/martin/Localhost/cloned/serenity/Tests/AK/TestComplex.cpp:9:
/Users/martin/Localhost/cloned/serenity/Meta/Lagom/../../AK/Complex.h:86:20: error: member reference base type 'const double' is not a structure or union
```

```
86 |         m_real += x.real();
    |                   ~^~~~~~
```

```
/Users/martin/Localhost/cloned/serenity/Tests/AK/TestComplex.cpp:152:20: error: member reference base type 'const double' is not a structure or union
```

```
152 |        x += a;
    |        ^
```

```
/Users/martin/Localhost/cloned/serenity/Tests/AK/TestComplex.cpp:166:20: error: member reference base type 'const double' is not a structure or union
```

```
166 |        auto c2 = c1 + r2;
    |                   ^
```

```
In file included from /Users/martin/Localhost/cloned/serenity/Tests/AK/TestComplex.cpp:9:
/Users/martin/Localhost/cloned/serenity/Meta/Lagom/../../AK/Complex.h:101:20: error: member reference base type 'const double' is not a structure or union
```

```
101 |        m_real -= x.real();
    |                   ~^~~~~~
```

```
/Users/martin/Localhost/cloned/serenity/Meta/Lagom/../../AK/Complex.h:168:11: note: in instantiation of function template specialization 'AK::Complex<double>::operator+<double>'
168 |         auto c2 = c1 + r2;
    |         ^
```

```
AK/Complex.h
12
@@ -83,7 +83,7 @@ class [[gnu::packed]] Complex {
83 83     template<AK::Concepts::Arithmetic U>
84 84     constexpr Complex<T> operator+=(U const& x)
85 85     {
86 86     -     m_real += x.real();
87 87     +     m_real += x;
88 88     return *this;
89 89     }

@@ -98,7 +98,7 @@ class [[gnu::packed]] Complex {
98 98     template<AK::Concepts::Arithmetic U>
99 99     constexpr Complex<T> operator-=(U const& x)
100 100    {
101 101    -     m_real -= x.real();
102 102    +     m_real -= x;
103 103    return *this;
104 104    }

@@ -224,31 +224,31 @@ class [[gnu::packed]] Complex {
```

```
function template specialization 'AK::
```

```
base type 'const int' is not a str
```

```
function template specialization 'AK::
```

```
base type 'const double' is not a s
```

```
of function template specialization
```

```
function template specialization 'AK:
```

```
base type 'const double' is not a
```

```

233 233
234 234     template<AK::Concepts::Arithmetic T, AK::Concepts::Arithmetic U>
235 - constexpr Complex<T> operator-(U const& b, Complex<T> const& a)
235 + constexpr Complex<T> operator-(U const& a, Complex<T> const& b)
236 236     {
237 237         Complex<T> x = a;
238 238         x -= b;
239 239         return x;
240 240     }
241 241

```

```

72
73 TEST_CASE(real_operators_regression)
74 {
75     {
76         auto c1 = Complex(1., 1.);
77         auto c2 = 1 - c1;
78         EXPECT_EQ(c2.real(), 0);
79         EXPECT_EQ(c2.imag(), -1);
80     }
81     {
82         auto c1 = Complex(1., 1.);
83         auto c2 = 1 / c1;
84         EXPECT_EQ(c2.real(), 0.5);
85         EXPECT_EQ(c2.imag(), -0.5);
86     }
87 }
88

```



```
vim
1 AK:
2 TestByteBuffer.cpp
3 TestChecked.cpp
4 TestCircularBuffer.cpp
5 TestCircularDeque.cpp
6 TestCircularQueue.cpp
7 TestComplex.cpp
8 TestDeprecatedString.cpp
9 TestDisjointChunks.cpp
10 TestDistinctNumeric.cpp
11 TestDoublyLinkedList.cpp
12 TestDuration.cpp
13 TestEnumBits.cpp
14 TestFind.cpp
15 TestFixedArray.cpp
16 TestFixedPoint.cpp
17 TestFloatingPoint.cpp
18 TestFloatingPointParsing.cpp
19 TestFlyString.cpp
20 TestFormat.cpp
21 TestFuzzyMatch.cpp
22 TestGenericLexer.cpp
23 TestHashFunctions.cpp
24 TestHashMap.cpp
25 TestHashTable.cpp
26 TestHex.cpp
27 TestIPv4Address.cpp
28 TestIPv6Address.cpp
29 TestIndexSequence.cpp
30 TestInsertionSort.cpp
31 TestIntegerMath.cpp
32 TestIntrusiveList.cpp
33 TestIntrusiveRedBlackTree.cpp
34 TestJSON.cpp
35 TestLEB128.cpp
36 TestLexicalPath.cpp
37 TestMACAddress.cpp
38 TestMemory.cpp
39 TestMemoryStream.cpp
40 TestNeverDestroyed.cpp
41 TestNonnullOwnPtr.cpp
42 TestNonnullRefPtr.cpp
43 TestNumberFormat.cpp
44 TestOptional.cpp
45 TestOwnPtr.cpp
46 TestPrint.cpp
47 TestQueue.cpp
48 TestQuickSelect.cpp
49 TestQuickSort.cpp
50 TestRedBlackTree.cpp
51 TestRefPtr.cpp
52 TestSIMD.cpp
53 TestSinglyLinkedList.cpp
54 TestSourceGenerator.cpp
55 TestSourceLocation.cpp
56 TestSpan.cpp
57 TestStack.cpp
58 TestStatistics.cpp
59 TestStdLibExtras.cpp
60 TestString.cpp
61 TestStringFloatingPointConversions.cpp
62 TestStringUtils.cpp
63 TestStringView.cpp
64 TestTrie.cpp
65 TestTuple.cpp
66 TestTypeTraits.cpp
67 TestTypedTransfer.cpp
68 TestUFixedBigInt.cpp
69 TestURL.cpp
70 TestUtf16.cpp
71 TestUtf8.cpp
72 TestVariant.cpp
73 TestVector.cpp
74 TestWeakPtr.cpp
75
76 Kernel:
77 TestEFault.cpp
78 TestEmptyPrivateInodeVMObject.cpp
79 TestEmptySharedInodeVMObject.cpp
80 TestExt2FS.cpp
81 TestInvalidUIDSet.cpp
82 TestKernelAlarm.cpp
83 TestKernelFilePermissions.cpp
84 TestKernelPledge.cpp
85 TestKernelUnveil.cpp
86 TestMemoryDeviceMmap.cpp
87 TestMunMap.cpp
88 TestPosixFallocate.cpp
89 TestPrivateInodeVMObject.cpp
90 TestProcFS.cpp
91 TestProcFSWrite.cpp
92 TestSharedInodeVMObject.cpp
93 TestSigAltStack.cpp
94 TestSigHandler.cpp
95 TestSigWait.cpp
96
97 LibAudio:
98 TestFLACSpec.cpp
99 TestPlaybackStream.cpp
100
101 LibC:
102 TestAbort.cpp
103 TestAssert.cpp
104 TestCType.cpp
105 TestEnvironment.cpp
106 TestIo.cpp
107 TestLibCDirEnt.cpp
108 TestLibCExec.cpp
109 TestLibCInodeWatcher.cpp
110 TestLibCMkTemp.cpp
111 TestLibCNetdb.cpp
112 TestLibCSetjmp.cpp
113 TestLibCString.cpp
114 TestLibCTime.cpp
115 TestMalloc.cpp
116 TestMath.cpp
117
118 LibCore:
119 TestLibCoreArgsParser.cpp
120 TestLibCoreDeferredInvoke.cpp
121 TestLibCoreFilePermissionsMa
122 TestLibCoreFileWatcher.cpp
123 TestLibCoreMappedFile.cpp
124 TestLibCorePromise.cpp
125 TestLibCoreSharedSingleProdu
126 TestLibCoreStream.cpp
127
128 LibCrypto:
129 TestAES.cpp
130 TestASN1.cpp
131 TestBigInteger.cpp
132 TestChaCha20.cpp
133 TestChacha20Poly1305.cpp
134 TestChecksum.cpp
135 TestCurves.cpp
136 TestEd25519.cpp
137 TestHMAC.cpp
138 TestHash.cpp
139 TestPBKDF2.cpp
140 TestStrncpy.cpp
141 TestStrtodAccuracy.cpp
142 TestWchar.cpp
143 TestWctype.cpp
144
145 LibC++:
146 test-cpp-parser.cpp
147 test-cpp-preprocessor.cpp
148
149 LibC++:
150 TestAES.cpp
151 TestASN1.cpp
152 TestBigInteger.cpp
153 TestChaCha20.cpp
154 TestChacha20Poly1305.cpp
155 TestChecksum.cpp
156 TestCurves.cpp
157 TestEd25519.cpp
158 TestHMAC.cpp
159 TestHash.cpp
160 TestPBKDF2.cpp
161
162 LibC++:
163 TestAES.cpp
164 TestASN1.cpp
165 TestBigInteger.cpp
166 TestChaCha20.cpp
167 TestChacha20Poly1305.cpp
168 TestChecksum.cpp
169 TestCurves.cpp
170 TestEd25519.cpp
171 TestHMAC.cpp
172 TestHash.cpp
173 TestPBKDF2.cpp
174
175 LibC++:
176 TestAES.cpp
177 TestASN1.cpp
178 TestBigInteger.cpp
179 TestChaCha20.cpp
180 TestChacha20Poly1305.cpp
181 TestChecksum.cpp
182 TestCurves.cpp
183 TestEd25519.cpp
184 TestHMAC.cpp
185 TestHash.cpp
186 TestPBKDF2.cpp
187
188 LibC++:
189 TestAES.cpp
190 TestASN1.cpp
191 TestBigInteger.cpp
192 TestChaCha20.cpp
193 TestChacha20Poly1305.cpp
194 TestChecksum.cpp
195 TestCurves.cpp
196 TestEd25519.cpp
197 TestHMAC.cpp
198 TestHash.cpp
199 TestPBKDF2.cpp
200
201 LibC++:
202 TestAES.cpp
203 TestASN1.cpp
204 TestBigInteger.cpp
205 TestChaCha20.cpp
206 TestChacha20Poly1305.cpp
207 TestChecksum.cpp
208 TestCurves.cpp
209 TestEd25519.cpp
210 TestHMAC.cpp
211 TestHash.cpp
212 TestPBKDF2.cpp
213
214 LibC++:
215 TestAES.cpp
216 TestASN1.cpp
217 TestBigInteger.cpp
218 TestChaCha20.cpp
219 TestChacha20Poly1305.cpp
220 TestChecksum.cpp
221 TestCurves.cpp
222 TestEd25519.cpp
223 TestHMAC.cpp
224 TestHash.cpp
225 TestPBKDF2.cpp
226
227 LibC++:
228 TestAES.cpp
229 TestASN1.cpp
230 TestBigInteger.cpp
231 TestChaCha20.cpp
232 TestChacha20Poly1305.cpp
233 TestChecksum.cpp
234 TestCurves.cpp
235 TestEd25519.cpp
236 TestHMAC.cpp
237 TestHash.cpp
238 TestPBKDF2.cpp
239
240 LibC++:
241 TestAES.cpp
242 TestASN1.cpp
243 TestBigInteger.cpp
244 TestChaCha20.cpp
245 TestChacha20Poly1305.cpp
246 TestChecksum.cpp
247 TestCurves.cpp
248 TestEd25519.cpp
249 TestHMAC.cpp
250 TestHash.cpp
251 TestPBKDF2.cpp
252
253 LibC++:
254 TestAES.cpp
255 TestASN1.cpp
256 TestBigInteger.cpp
257 TestChaCha20.cpp
258 TestChacha20Poly1305.cpp
259 TestChecksum.cpp
260 TestCurves.cpp
261 TestEd25519.cpp
262 TestHMAC.cpp
263 TestHash.cpp
264 TestPBKDF2.cpp
265
266 LibC++:
267 TestAES.cpp
268 TestASN1.cpp
269 TestBigInteger.cpp
270 TestChaCha20.cpp
271 TestChacha20Poly1305.cpp
272 TestChecksum.cpp
273 TestCurves.cpp
274 TestEd25519.cpp
275 TestHMAC.cpp
276 TestHash.cpp
277 TestPBKDF2.cpp
278
279 LibC++:
280 TestAES.cpp
281 TestASN1.cpp
282 TestBigInteger.cpp
283 TestChaCha20.cpp
284 TestChacha20Poly1305.cpp
285 TestChecksum.cpp
286 TestCurves.cpp
287 TestEd25519.cpp
288 TestHMAC.cpp
289 TestHash.cpp
290 TestPBKDF2.cpp
291
292 LibC++:
293 TestAES.cpp
294 TestASN1.cpp
295 TestBigInteger.cpp
296 TestChaCha20.cpp
297 TestChacha20Poly1305.cpp
298 TestChecksum.cpp
299 TestCurves.cpp
300 TestEd25519.cpp
301 TestHMAC.cpp
302 TestHash.cpp
303 TestPBKDF2.cpp
304
305 LibC++:
306 TestAES.cpp
307 TestASN1.cpp
308 TestBigInteger.cpp
309 TestChaCha20.cpp
310 TestChacha20Poly1305.cpp
311 TestChecksum.cpp
312 TestCurves.cpp
313 TestEd25519.cpp
314 TestHMAC.cpp
315 TestHash.cpp
316 TestPBKDF2.cpp
317
318 LibC++:
319 TestAES.cpp
320 TestASN1.cpp
321 TestBigInteger.cpp
322 TestChaCha20.cpp
323 TestChacha20Poly1305.cpp
324 TestChecksum.cpp
325 TestCurves.cpp
326 TestEd25519.cpp
327 TestHMAC.cpp
328 TestHash.cpp
329 TestPBKDF2.cpp
330
331 LibC++:
332 TestAES.cpp
333 TestASN1.cpp
334 TestBigInteger.cpp
335 TestChaCha20.cpp
336 TestChacha20Poly1305.cpp
337 TestChecksum.cpp
338 TestCurves.cpp
339 TestEd25519.cpp
340 TestHMAC.cpp
341 TestHash.cpp
342 TestPBKDF2.cpp
343
344 LibC++:
345 TestAES.cpp
346 TestASN1.cpp
347 TestBigInteger.cpp
348 TestChaCha20.cpp
349 TestChacha20Poly1305.cpp
350 TestChecksum.cpp
351 TestCurves.cpp
352 TestEd25519.cpp
353 TestHMAC.cpp
354 TestHash.cpp
355 TestPBKDF2.cpp
356
357 LibC++:
358 TestAES.cpp
359 TestASN1.cpp
360 TestBigInteger.cpp
361 TestChaCha20.cpp
362 TestChacha20Poly1305.cpp
363 TestChecksum.cpp
364 TestCurves.cpp
365 TestEd25519.cpp
366 TestHMAC.cpp
367 TestHash.cpp
368 TestPBKDF2.cpp
369
370 LibC++:
371 TestAES.cpp
372 TestASN1.cpp
373 TestBigInteger.cpp
374 TestChaCha20.cpp
375 TestChacha20Poly1305.cpp
376 TestChecksum.cpp
377 TestCurves.cpp
378 TestEd25519.cpp
379 TestHMAC.cpp
380 TestHash.cpp
381 TestPBKDF2.cpp
382
383 LibC++:
384 TestAES.cpp
385 TestASN1.cpp
386 TestBigInteger.cpp
387 TestChaCha20.cpp
388 TestChacha20Poly1305.cpp
389 TestChecksum.cpp
390 TestCurves.cpp
391 TestEd25519.cpp
392 TestHMAC.cpp
393 TestHash.cpp
394 TestPBKDF2.cpp
395
396 LibC++:
397 TestAES.cpp
398 TestASN1.cpp
399 TestBigInteger.cpp
400 TestChaCha20.cpp
401 TestChacha20Poly1305.cpp
402 TestChecksum.cpp
403 TestCurves.cpp
404 TestEd25519.cpp
405 TestHMAC.cpp
406 TestHash.cpp
407 TestPBKDF2.cpp
408
409 LibC++:
410 TestAES.cpp
411 TestASN1.cpp
412 TestBigInteger.cpp
413 TestChaCha20.cpp
414 TestChacha20Poly1305.cpp
415 TestChecksum.cpp
416 TestCurves.cpp
417 TestEd25519.cpp
418 TestHMAC.cpp
419 TestHash.cpp
420 TestPBKDF2.cpp
421
422 LibC++:
423 TestAES.cpp
424 TestASN1.cpp
425 TestBigInteger.cpp
426 TestChaCha20.cpp
427 TestChacha20Poly1305.cpp
428 TestChecksum.cpp
429 TestCurves.cpp
430 TestEd25519.cpp
431 TestHMAC.cpp
432 TestHash.cpp
433 TestPBKDF2.cpp
434
435 LibC++:
436 TestAES.cpp
437 TestASN1.cpp
438 TestBigInteger.cpp
439 TestChaCha20.cpp
440 TestChacha20Poly1305.cpp
441 TestChecksum.cpp
442 TestCurves.cpp
443 TestEd25519.cpp
444 TestHMAC.cpp
445 TestHash.cpp
446 TestPBKDF2.cpp
447
448 LibC++:
449 TestAES.cpp
450 TestASN1.cpp
451 TestBigInteger.cpp
452 TestChaCha20.cpp
453 TestChacha20Poly1305.cpp
454 TestChecksum.cpp
455 TestCurves.cpp
456 TestEd25519.cpp
457 TestHMAC.cpp
458 TestHash.cpp
459 TestPBKDF2.cpp
460
461 LibC++:
462 TestAES.cpp
463 TestASN1.cpp
464 TestBigInteger.cpp
465 TestChaCha20.cpp
466 TestChacha20Poly1305.cpp
467 TestChecksum.cpp
468 TestCurves.cpp
469 TestEd25519.cpp
470 TestHMAC.cpp
471 TestHash.cpp
472 TestPBKDF2.cpp
473
474 LibC++:
475 TestAES.cpp
476 TestASN1.cpp
477 TestBigInteger.cpp
478 TestChaCha20.cpp
479 TestChacha20Poly1305.cpp
480 TestChecksum.cpp
481 TestCurves.cpp
482 TestEd25519.cpp
483 TestHMAC.cpp
484 TestHash.cpp
485 TestPBKDF2.cpp
486
487 LibC++:
488 TestAES.cpp
489 TestASN1.cpp
490 TestBigInteger.cpp
491 TestChaCha20.cpp
492 TestChacha20Poly1305.cpp
493 TestChecksum.cpp
494 TestCurves.cpp
495 TestEd25519.cpp
496 TestHMAC.cpp
497 TestHash.cpp
498 TestPBKDF2.cpp
499
500 LibC++:
501 TestAES.cpp
502 TestASN1.cpp
503 TestBigInteger.cpp
504 TestChaCha20.cpp
505 TestChacha20Poly1305.cpp
506 TestChecksum.cpp
507 TestCurves.cpp
508 TestEd25519.cpp
509 TestHMAC.cpp
510 TestHash.cpp
511 TestPBKDF2.cpp
512
513 LibC++:
514 TestAES.cpp
515 TestASN1.cpp
516 TestBigInteger.cpp
517 TestChaCha20.cpp
518 TestChacha20Poly1305.cpp
519 TestChecksum.cpp
520 TestCurves.cpp
521 TestEd25519.cpp
522 TestHMAC.cpp
523 TestHash.cpp
524 TestPBKDF2.cpp
525
526 LibC++:
527 TestAES.cpp
528 TestASN1.cpp
529 TestBigInteger.cpp
530 TestChaCha20.cpp
531 TestChacha20Poly1305.cpp
532 TestChecksum.cpp
533 TestCurves.cpp
534 TestEd25519.cpp
535 TestHMAC.cpp
536 TestHash.cpp
537 TestPBKDF2.cpp
538
539 LibC++:
540 TestAES.cpp
541 TestASN1.cpp
542 TestBigInteger.cpp
543 TestChaCha20.cpp
544 TestChacha20Poly1305.cpp
545 TestChecksum.cpp
546 TestCurves.cpp
547 TestEd25519.cpp
548 TestHMAC.cpp
549 TestHash.cpp
550 TestPBKDF2.cpp
551
552 LibC++:
553 TestAES.cpp
554 TestASN1.cpp
555 TestBigInteger.cpp
556 TestChaCha20.cpp
557 TestChacha20Poly1305.cpp
558 TestChecksum.cpp
559 TestCurves.cpp
560 TestEd25519.cpp
561 TestHMAC.cpp
562 TestHash.cpp
563 TestPBKDF2.cpp
564
565 LibC++:
566 TestAES.cpp
567 TestASN1.cpp
568 TestBigInteger.cpp
569 TestChaCha20.cpp
570 TestChacha20Poly1305.cpp
571 TestChecksum.cpp
572 TestCurves.cpp
573 TestEd25519.cpp
574 TestHMAC.cpp
575 TestHash.cpp
576 TestPBKDF2.cpp
577
578 LibC++:
579 TestAES.cpp
580 TestASN1.cpp
581 TestBigInteger.cpp
582 TestChaCha20.cpp
583 TestChacha20Poly1305.cpp
584 TestChecksum.cpp
585 TestCurves.cpp
586 TestEd25519.cpp
587 TestHMAC.cpp
588 TestHash.cpp
589 TestPBKDF2.cpp
590
591 LibC++:
592 TestAES.cpp
593 TestASN1.cpp
594 TestBigInteger.cpp
595 TestChaCha20.cpp
596 TestChacha20Poly1305.cpp
597 TestChecksum.cpp
598 TestCurves.cpp
599 TestEd25519.cpp
600 TestHMAC.cpp
601 TestHash.cpp
602 TestPBKDF2.cpp
603
604 LibC++:
605 TestAES.cpp
606 TestASN1.cpp
607 TestBigInteger.cpp
608 TestChaCha20.cpp
609 TestChacha20Poly1305.cpp
610 TestChecksum.cpp
611 TestCurves.cpp
612 TestEd25519.cpp
613 TestHMAC.cpp
614 TestHash.cpp
615 TestPBKDF2.cpp
616
617 LibC++:
618 TestAES.cpp
619 TestASN1.cpp
620 TestBigInteger.cpp
621 TestChaCha20.cpp
622 TestChacha20Poly1305.cpp
623 TestChecksum.cpp
624 TestCurves.cpp
625 TestEd25519.cpp
626 TestHMAC.cpp
627 TestHash.cpp
628 TestPBKDF2.cpp
629
630 LibC++:
631 TestAES.cpp
632 TestASN1.cpp
633 TestBigInteger.cpp
634 TestChaCha20.cpp
635 TestChacha20Poly1305.cpp
636 TestChecksum.cpp
637 TestCurves.cpp
638 TestEd25519.cpp
639 TestHMAC.cpp
640 TestHash.cpp
641 TestPBKDF2.cpp
642
643 LibC++:
644 TestAES.cpp
645 TestASN1.cpp
646 TestBigInteger.cpp
647 TestChaCha20.cpp
648 TestChacha20Poly1305.cpp
649 TestChecksum.cpp
650 TestCurves.cpp
651 TestEd25519.cpp
652 TestHMAC.cpp
653 TestHash.cpp
654 TestPBKDF2.cpp
655
656 LibC++:
657 TestAES.cpp
658 TestASN1.cpp
659 TestBigInteger.cpp
660 TestChaCha20.cpp
661 TestChacha20Poly1305.cpp
662 TestChecksum.cpp
663 TestCurves.cpp
664 TestEd25519.cpp
665 TestHMAC.cpp
666 TestHash.cpp
667 TestPBKDF2.cpp
668
669 LibC++:
670 TestAES.cpp
671 TestASN1.cpp
672 TestBigInteger.cpp
673 TestChaCha20.cpp
674 TestChacha20Poly1305.cpp
675 TestChecksum.cpp
676 TestCurves.cpp
677 TestEd25519.cpp
678 TestHMAC.cpp
679 TestHash.cpp
680 TestPBKDF2.cpp
681
682 LibC++:
683 TestAES.cpp
684 TestASN1.cpp
685 TestBigInteger.cpp
686 TestChaCha20.cpp
687 TestChacha20Poly1305.cpp
688 TestChecksum.cpp
689 TestCurves.cpp
690 TestEd25519.cpp
691 TestHMAC.cpp
692 TestHash.cpp
693 TestPBKDF2.cpp
694
695 LibC++:
696 TestAES.cpp
697 TestASN1.cpp
698 TestBigInteger.cpp
699 TestChaCha20.cpp
700 TestChacha20Poly1305.cpp
701 TestChecksum.cpp
702 TestCurves.cpp
703 TestEd25519.cpp
704 TestHMAC.cpp
705 TestHash.cpp
706 TestPBKDF2.cpp
707
708 LibC++:
709 TestAES.cpp
710 TestASN1.cpp
711 TestBigInteger.cpp
712 TestChaCha20.cpp
713 TestChacha20Poly1305.cpp
714 TestChecksum.cpp
715 TestCurves.cpp
716 TestEd25519.cpp
717 TestHMAC.cpp
718 TestHash.cpp
719 TestPBKDF2.cpp
720
721 LibC++:
722 TestAES.cpp
723 TestASN1.cpp
724 TestBigInteger.cpp
725 TestChaCha20.cpp
726 TestChacha20Poly1305.cpp
727 TestChecksum.cpp
728 TestCurves.cpp
729 TestEd25519.cpp
730 TestHMAC.cpp
731 TestHash.cpp
732 TestPBKDF2.cpp
733
734 LibC++:
735 TestAES.cpp
736 TestASN1.cpp
737 TestBigInteger.cpp
738 TestChaCha20.cpp
739 TestChacha20Poly1305.cpp
740 TestChecksum.cpp
741 TestCurves.cpp
742 TestEd25519.cpp
743 TestHMAC.cpp
744 TestHash.cpp
745 TestPBKDF2.cpp
746
747 LibC++:
748 TestAES.cpp
749 TestASN1.cpp
750 TestBigInteger.cpp
751 TestChaCha20.cpp
752 TestChacha20Poly1305.cpp
753 TestChecksum.cpp
754 TestCurves.cpp
755 TestEd25519.cpp
756 TestHMAC.cpp
757 TestHash.cpp
758 TestPBKDF2.cpp
759
760 LibC++:
761 TestAES.cpp
762 TestASN1.cpp
763 TestBigInteger.cpp
764 TestChaCha20.cpp
765 TestChacha20Poly1305.cpp
766 TestChecksum.cpp
767 TestCurves.cpp
768 TestEd25519.cpp
769 TestHMAC.cpp
770 TestHash.cpp
771 TestPBKDF2.cpp
772
773 LibC++:
774 TestAES.cpp
775 TestASN1.cpp
776 TestBigInteger.cpp
777 TestChaCha20.cpp
778 TestChacha20Poly1305.cpp
779 TestChecksum.cpp
780 TestCurves.cpp
781 TestEd25519.cpp
782 TestHMAC.cpp
783 TestHash.cpp
784 TestPBKDF2.cpp
785
786 LibC++:
787 TestAES.cpp
788 TestASN1.cpp
789 TestBigInteger.cpp
790 TestChaCha20.cpp
791 TestChacha20Poly1305.cpp
792 TestChecksum.cpp
793 TestCurves.cpp
794 TestEd25519.cpp
795 TestHMAC.cpp
796 TestHash.cpp
797 TestPBKDF2.cpp
798
799 LibC++:
800 TestAES.cpp
801 TestASN1.cpp
802 TestBigInteger.cpp
803 TestChaCha20.cpp
804 TestChacha20Poly1305.cpp
805 TestChecksum.cpp
806 TestCurves.cpp
807 TestEd25519.cpp
808 TestHMAC.cpp
809 TestHash.cpp
810 TestPBKDF2.cpp
811
812 LibC++:
813 TestAES.cpp
814 TestASN1.cpp
815 TestBigInteger.cpp
816 TestChaCha20.cpp
817 TestChacha20Poly1305.cpp
818 TestChecksum.cpp
819 TestCurves.cpp
820 TestEd25519.cpp
821 TestHMAC.cpp
822 TestHash.cpp
823 TestPBKDF2.cpp
824
825 LibC++:
826 TestAES.cpp
827 TestASN1.cpp
828 TestBigInteger.cpp
829 TestChaCha20.cpp
830 TestChacha20Poly1305.cpp
831 TestChecksum.cpp
832 TestCurves.cpp
833 TestEd25519.cpp
834 TestHMAC.cpp
835 TestHash.cpp
836 TestPBKDF2.cpp
837
838 LibC++:
839 TestAES.cpp
840 TestASN1.cpp
841 TestBigInteger.cpp
842 TestChaCha20.cpp
843 TestChacha20Poly1305.cpp
844 TestChecksum.cpp
845 TestCurves.cpp
846 TestEd25519.cpp
847 TestHMAC.cpp
848 TestHash.cpp
849 TestPBKDF2.cpp
850
851 LibC++:
852 TestAES.cpp
853 TestASN1.cpp
854 TestBigInteger.cpp
855 TestChaCha20.cpp
856 TestChacha20Poly1305.cpp
857 TestChecksum.cpp
858 TestCurves.cpp
859 TestEd25519.cpp
860 TestHMAC.cpp
861 TestHash.cpp
862 TestPBKDF2.cpp
863
864 LibC++:
865 TestAES.cpp
866 TestASN1.cpp
867 TestBigInteger.cpp
868 TestChaCha20.cpp
869 TestChacha20Poly1305.cpp
870 TestChecksum.cpp
871 TestCurves.cpp
872 TestEd25519.cpp
873 TestHMAC.cpp
874 TestHash.cpp
875 TestPBKDF2.cpp
876
877 LibC++:
878 TestAES.cpp
879 TestASN1.cpp
880 TestBigInteger.cpp
881 TestChaCha20.cpp
882 TestChacha20Poly1305.cpp
883 TestChecksum.cpp
884 TestCurves.cpp
885 TestEd25519.cpp
886 TestHMAC.cpp
887 TestHash.cpp
888 TestPBKDF2.cpp
889
890 LibC++:
891 TestAES.cpp
892 TestASN1.cpp
893 TestBigInteger.cpp
894 TestChaCha20.cpp
895 TestChacha20Poly1305.cpp
896 TestChecksum.cpp
897 TestCurves.cpp
898 TestEd25519.cpp
899 TestHMAC.cpp
900 TestHash.cpp
901 TestPBKDF2.cpp
902
903 LibC++:
904 TestAES.cpp
905 TestASN1.cpp
906 TestBigInteger.cpp
907 TestChaCha20.cpp
908 TestChacha20Poly1305.cpp
909 TestChecksum.cpp
910 TestCurves.cpp
911 TestEd25519.cpp
912 TestHMAC.cpp
913 TestHash.cpp
914 TestPBKDF2.cpp
915
916 LibC++:
917 TestAES.cpp
918 TestASN1.cpp
919 TestBigInteger.cpp
920 TestChaCha20.cpp
921 TestChacha20Poly1305.cpp
922 TestChecksum.cpp
923 TestCurves.cpp
924 TestEd25519.cpp
925 TestHMAC.cpp
926 TestHash.cpp
927 TestPBKDF2.cpp
928
929 LibC++:
930 TestAES.cpp
931 TestASN1.cpp
932 TestBigInteger.cpp
933 TestChaCha20.cpp
934 TestChacha20Poly1305.cpp
935 TestChecksum.cpp
936 TestCurves.cpp
937 TestEd25519.cpp
938 TestHMAC.cpp
939 TestHash.cpp
940 TestPBKDF2.cpp
941
942 LibC++:
943 TestAES.cpp
944 TestASN1.cpp
945 TestBigInteger.cpp
946 TestChaCha20.cpp
947 TestChacha20Poly1305.cpp
948 TestChecksum.cpp
949 TestCurves.cpp
950 TestEd25519.cpp
951 TestHMAC.cpp
952 TestHash.cpp
953 TestPBKDF2.cpp
954
955 LibC++:
956 TestAES.cpp
957 TestASN1.cpp
958 TestBigInteger.cpp
959 TestChaCha20.cpp
960 TestChacha20Poly1305.cpp
961 TestChecksum.cpp
962 TestCurves.cpp
963 TestEd25519.cpp
964 TestHMAC.cpp
965 TestHash.cpp
966 TestPBKDF2.cpp
967
968 LibC++:
969 TestAES.cpp
970 TestASN1.cpp
971 TestBigInteger.cpp
972 TestChaCha20.cpp
973 TestChacha20Poly1305.cpp
974 TestChecksum.cpp
975 TestCurves.cpp
976 TestEd25519.cpp
977 TestHMAC.cpp
978 TestHash.cpp
979 TestPBKDF2.cpp
980
981 LibC++:
982 TestAES.cpp
983 TestASN1.cpp
984 TestBigInteger.cpp
985 TestChaCha20.cpp
986 TestChacha20Poly1305.cpp
987 TestChecksum.cpp
988 TestCurves.cpp
989 TestEd25519.cpp
990 TestHMAC.cpp
991 TestHash.cpp
992 TestPBKDF2.cpp
993
994 LibC++:
995 TestAES.cpp
996 TestASN1.cpp
997 TestBigInteger.cpp
998 TestChaCha20.cpp
999 TestChacha20Poly1305.cpp
1000 TestChecksum.cpp
1001 TestCurves.cpp
1002 TestEd25519.cpp
1003 TestHMAC.cpp
1004 TestHash.cpp
1005 TestPBKDF2.cpp
1006
1007 LibC++:
1008 TestAES.cpp
1009 TestASN1.cpp
1010 TestBigInteger.cpp
1011 TestChaCha20.cpp
1012 TestChacha20Poly1305.cpp
1013 TestChecksum.cpp
1014 TestCurves.cpp
1015 TestEd25519.cpp
1016 TestHMAC.cpp
1017 TestHash.cpp
1018 TestPBKDF2.cpp
1019
1020 LibC++:
1021 TestAES.cpp
1022 TestASN1.cpp
1023 TestBigInteger.cpp
1024 TestChaCha20.cpp
1025 TestChacha20Poly1305.cpp
1026 TestChecksum.cpp
1027 TestCurves.cpp
1028 TestEd25519.cpp
1029 TestHMAC.cpp
1030 TestHash.cpp
1031 TestPBKDF2.cpp
1032
1033 LibC++:
1034 TestAES.cpp
1035 TestASN1.cpp
1036 TestBigInteger.cpp
1037 TestChaCha20.cpp
1038 TestChacha20Poly1305.cpp
1039 TestChecksum.cpp
1040 TestCurves.cpp
1041 TestEd25519.cpp
1042 TestHMAC.cpp
1043 TestHash.cpp
1044 TestPBKDF2.cpp
1045
1046 LibC++:
1047 TestAES.cpp
1048 TestASN1.cpp
1049 TestBigInteger.cpp
1050 TestChaCha20.cpp
1051 TestChacha20Poly1305.cpp
1052 TestChecksum.cpp
1053 TestCurves.cpp
1054 TestEd25519.cpp
1055 TestHMAC.cpp
1056 TestHash.cpp
1057 TestPBKDF2.cpp
1058
1059 LibC++:
1060 TestAES.cpp
1061 TestASN1.cpp
1062 TestBigInteger.cpp
1063 TestChaCha20.cpp
1064 TestChacha20Poly1305.cpp
1065 TestChecksum.cpp
1066 TestCurves.cpp
1067 TestEd25519.cpp
1068 TestHMAC.cpp
1069 TestHash.cpp
1070 TestPBKDF2.cpp
1071
1072 LibC++:
1073 TestAES.cpp
1074 TestASN1.cpp
1075 TestBigInteger.cpp
1076 TestChaCha20.cpp
1077 TestChacha20Poly1305.cpp
1078 TestChecksum.cpp
1079 TestCurves.cpp
1080 TestEd25519.cpp
1081 TestHMAC.cpp
1082 TestHash.cpp
1083 TestPBKDF2.cpp
1084
1085 LibC++:
1086 TestAES.cpp
1087 TestASN1.cpp
1088 TestBigInteger.cpp
1089 TestChaCha20.cpp
1090 TestChacha20Poly1305.cpp
1091 TestChecksum.cpp
1092 TestCurves.cpp
1093 TestEd25519.cpp
1094 TestHMAC.cpp
1095 TestHash.cpp
1096 TestPBKDF2.cpp
1097
1098 LibC++:
1099 TestAES.cpp
1100 TestASN1.cpp
1101 TestBigInteger.cpp
1102 TestChaCha20.cpp
1103 TestChacha20Poly1305.cpp
1104 TestChecksum.cpp
1105 TestCurves.cpp
1106 TestEd25519.cpp
1107 TestHMAC.cpp
1108 TestHash.cpp
1109 TestPBKDF2.cpp
1110
1111 LibC++:
1112 TestAES.cpp
1113 TestASN1.cpp
1114 TestBigInteger.cpp
1115 TestChaCha20.cpp
1116 TestChacha20Poly1305.cpp
1117 TestChecksum.cpp
1118 TestCurves.cpp
1119 TestEd25519.cpp
1120 TestHMAC.cpp
1121 TestHash.cpp
1122 TestPBKDF2.cpp
1123
1124 LibC++:
1125 TestAES.cpp
1126 TestASN1.cpp
1127 TestBigInteger.cpp
1128 TestChaCha20.cpp
1129 TestChacha20Poly1305.cpp
1130 TestChecksum.cpp
1131 TestCurves.cpp
1132 TestEd25519.cpp
1133 TestHMAC.cpp
1134 TestHash.cpp
1135 TestPBKDF2.cpp
1136
1137 LibC++:
1138 TestAES.cpp
1139 TestASN1.cpp
1140 TestBigInteger.cpp
1141 TestChaCha20.cpp
1142 TestChacha20Poly1305.cpp
1143 TestChecksum.cpp
1144 TestCurves.cpp
1145 TestEd25519.cpp
1146 TestHMAC.cpp
1147 TestHash.cpp
1148 TestPBKDF2.cpp
1149
1150 LibC++:
1151 TestAES.cpp
1152 TestASN1.cpp
1153 TestBigInteger.cpp
1154 TestChaCha20.cpp
1155 TestChacha20Poly1305.cpp
1156 TestChecksum.cpp
1157 TestCurves.cpp
1158 TestEd25519.cpp
1159 TestHMAC.cpp
1160 TestHash.cpp
1161 TestPBKDF2.cpp
1162
1163 LibC++:
1164 TestAES.cpp
1165 TestASN1.cpp
1166 TestBigInteger.cpp
1167 TestChaCha20.cpp
1168 TestChacha20Poly1305.cpp
1169 TestChecksum.cpp
1170 TestCurves.cpp
1171 TestEd25519.cpp
1172 TestHMAC.cpp
1173 TestHash.cpp
1174 TestPBKDF2.cpp
1175
1176 LibC++:
1177 TestAES.cpp
1178 TestASN1.cpp
1179 TestBigInteger.cpp
1180 TestChaCha20.cpp
1181 TestChacha20Poly1305.cpp
1182 TestChecksum.cpp
1183 TestCurves.cpp
1184 TestEd25519.cpp
1185 TestHMAC.cpp
1186 TestHash.cpp
1187 TestPBKDF2.cpp
1188
1189 LibC++:
1190 TestAES.cpp
1191 TestASN1.cpp
1192 TestBigInteger.cpp
1193 TestChaCha20.cpp
1194 TestChacha20Poly1305.cpp
1195 TestChecksum.cpp
1196 TestCurves.cpp
1197 TestEd25519.cpp
1198 TestHMAC.cpp
1199 TestHash.cpp
1200 TestPBKDF2.cpp
1201
1202 LibC++:
1203 TestAES.cpp
1204 TestASN1.cpp
1205 TestBigInteger.cpp
1206 TestChaCha20.cpp
1207 TestChacha20Poly1305.cpp
1208 TestChecksum.cpp
1209 TestCurves.cpp
1210 TestEd25519.cpp
1211 TestHMAC.cpp
1212 TestHash.cpp
1213 TestPBKDF2.cpp
1214
1215 LibC++:
1216 TestAES.cpp
1217 TestASN1.cpp
1218 TestBigInteger.cpp
1219 TestChaCha20.cpp
1220 TestChacha20Poly1305.cpp
1221 TestChecksum.cpp
1222 TestCurves.cpp
1223 TestEd25519.cpp
1224 TestHMAC.cpp
1225 TestHash.cpp
1226 TestPBKDF2.cpp
1227
1228 LibC++:
1229 TestAES.cpp
1230 TestASN1.cpp
1231 TestBigInteger.cpp
1232 TestChaCha20.cpp
1233 TestChacha20Poly1305.cpp
1234 TestChecksum.cpp
1235 TestCurves.cpp
1236 TestEd25519.cpp
1237 TestHMAC.cpp
1238 TestHash.cpp
1239 TestPBKDF2.cpp
1240
1241 LibC++:
1242 TestAES.cpp
1243 TestASN1.cpp
1244 TestBigInteger.cpp
1245 TestChaCha20.cpp
1246 TestChacha20Poly1305.cpp
1247 TestChecksum.cpp
1248 TestCurves.cpp
1249 TestEd25519.cpp
1250 TestHMAC.cpp
1251 TestHash.cpp
1252 TestPBKDF2.cpp
1253
1254 LibC++:
1255 TestAES.cpp
1256 TestASN1.cpp
1257 TestBigInteger.cpp
1258 TestChaCha20.cpp
1259 TestChacha20Poly1305.cpp
1260 TestChecksum.cpp
1261 TestCurves.cpp
1262 TestEd25519.cpp
1263 TestHMAC.cpp
1264 TestHash.cpp
1265 TestPBKDF2.cpp
1266
1267 LibC++:
1268 TestAES.cpp
1269 TestASN1.cpp
1270 TestBigInteger.cpp
1271 TestChaCha20.cpp
1272 TestChacha20Poly1305.cpp
1273 TestChecksum.cpp
1274 TestCurves.cpp
1275 TestEd25519.cpp
1276 TestHMAC.cpp
1277 TestHash.cpp
1278 TestPBKDF2.cpp
1279
1280 LibC++:
1281 TestAES.cpp
1282 TestASN1.cpp
1283 TestBigInteger.cpp
1284 TestChaCha20.cpp
1285 TestChacha20Poly1305.cpp
1286 TestChecksum.cpp
1287 TestCurves.cpp
1288 TestEd25519.cpp
1289 TestHMAC.cpp
1290 TestHash.cpp
1291 TestPBKDF2.cpp
1292
1293 LibC++:
1294 TestAES.cpp
1295 TestASN1.cpp
1296 TestBigInteger.cpp
1297 TestChaCha20.cpp
1298 TestChacha20Poly1305.cpp
1299 TestChecksum.cpp
1300 TestCurves.cpp
1301 TestEd25519.cpp
1302 TestHMAC.cpp
1303 TestHash.cpp
1304 TestPBKDF2.cpp
1305
1306 LibC++:
1307 TestAES.cpp
1308 TestASN1.cpp
1309 TestBigInteger.cpp
1310 TestChaCha20.cpp
1311 TestChacha20Poly1305.cpp
1312 TestChecksum.cpp
1313 TestCurves.cpp
1314 TestEd25519.cpp
1315 TestHMAC.cpp
1316 TestHash.cpp
1317 TestPBKDF2.cpp
1318
1319 LibC++:
1320 TestAES.cpp
1321 TestASN1.cpp
1322 TestBigInteger.cpp
1323 TestChaCha20.cpp
1324 TestChacha20Poly1305.cpp
1325 TestChecksum.cpp
1326 TestCurves.cpp
1327 TestEd25519.cpp
1328 TestHMAC.cpp
1329 TestHash.cpp
1330 TestPBKDF2.cpp
1331
1332 LibC++:
1333 TestAES.cpp
1334 TestASN1.cpp
1335 TestBigInteger.cpp
1336 TestChaCha20.cpp
1337 TestChacha20Poly1305.cpp
1338 TestChecksum.cpp
1339 TestCurves.cpp
1340 TestEd25519.cpp
1341 TestHMAC.cpp
1342 TestHash.cpp
1343 TestPBKDF2.cpp
1344
1345 LibC++:
1346 TestAES.cpp
1347 TestASN1.cpp
1348 TestBigInteger.cpp
1349 TestChaCha20.cpp
1350 TestChacha20Poly1305.cpp
1351 TestChecksum.cpp
1352 TestCurves.cpp
1353 TestEd25519.cpp
1354 TestHMAC.cpp
1355 TestHash.cpp
1356 TestPBKDF2.cpp
1357
1358 LibC++:
1359 TestAES.cpp
1360 TestASN1.cpp
1361 TestBigInteger.cpp
1362 TestChaCha20.cpp
1363 TestChacha20Poly1305.cpp
1364 TestChecksum.cpp
1365 TestCurves.cpp
1366 TestEd25519.cpp
1367 TestHMAC.cpp
1368 TestHash.cpp
1369 TestPBKDF2.cpp
1370
1371 LibC++:
1372 TestAES.cpp
1373 TestASN1.cpp
1374 TestBigInteger.cpp
1375 TestChaCha20.cpp
1376 TestChacha20Poly1305.cpp
1377 TestChecksum.cpp
1378 TestCurves.cpp
1379 TestEd25519.cpp
1380 TestHMAC.cpp
1381 TestHash.cpp
1382 TestPBKDF2.cpp
1383
1384 LibC++:
1385 TestAES.cpp
1386 TestASN1.cpp
1387 TestBigInteger.cpp
1388 TestChaCha20.cpp
1389 TestChacha20Poly1305.cpp
1390 TestChecksum.cpp
1391 TestCurves.cpp
1392 TestEd25519.cpp
1393 TestHMAC.cpp
1394 TestHash.cpp
1395 TestPBKDF2.cpp
1396
1397 LibC++:
1398 TestAES.cpp
1399 TestASN1.cpp
1400 TestBigInteger.cpp
1401 TestChaCha20.cpp
1402 TestChacha20Poly1305.cpp
1403 TestChecksum.cpp
1404 TestCurves.cpp
1405 TestEd25519.cpp
1
```


Summary

Summary

- Property tests are great!

Summary

- Property tests are great!
- For the love of God, steal from Hypothesis

Summary

- Property tests are great!
- For the love of God, steal from Hypothesis
- Functional C++ is hard; don't go against the grain

Summary

- Property tests are great!
- For the love of God, steal from Hypothesis
- Functional C++ is hard; don't go against the grain
- SerenityOS+PBT is a learning opportunity



Thank you!

@janiczek