Experiment 2: helloworld

Coding is needed. But no code submission is required.

Convert to PDF before uploading.

Related project description:

https://fxlin.github.io/p3-tee/quickerstart/

https://fxlin.github.io/p3-tee/helloworld/

Context & Sessions (20%)

Use 1-2 sentences for each question below.

1. What is a TEE "context"? Explain the purpose of this object.

   The TEE Context are containers for sessions and their purpose is to create a logical connection between the client and the TEE.

2. In one CA, how many TEE contexts can be simultaneously alive (i.e. initialized but not finalized)? You can determine this with a small experiment, e.g. keep creating TEE contexts until the creation fails.

   It seems like you can only have 1022 TEE context alive at a time.

3. In one context, how many sessions can be opened simultaneously? You can determine this with a small experiment, e.g. keep creating TEE sessions in the same context until the creation fails.

   Only 17 sessions can be created per one session.

You may need to write some experiment code to validate your answers above. Just state your observations.

Change `helloworld` (40%)

The code description: https://fxlin.github.io/p3-tee/helloworld/#app-1-helloworld

Change the source code: if the resultant value (i.e. after incrementing or decrementing) is NOT in the range of [0, 100], TA should return an error to the CA.

How did you implement the above logic? In space below, paste the **key** lines you added/modified, which should not be more than 20 lines (10%)

```
    //checks if value is greater than 100 or less than 0
    if (params[0].value.a > 100 || params[0].value.a < 0) {
            return -100;
    }
```

Implemented in static TEE_Result inc_value(uint32_t, param_types, TEE_Param params[4]) and static TEE_Result dec_value(uint32_t param_types, TEE_Param params[4]) in ta/hello_world_ta.c.

```
if (res == -100) {
        errx(1, "Value is %d which is out of bounds of [0, 100]", op.params[0].value.a);
}
```

Implemented in int main(void) in host/main.c.

How do you verify that your code works properly? E.g. what values does CA send to TA? Does your CA print an error code returned by the TA? Attach a screenshot to prove it. (30%)

The CA will send parameters to the TA and the TA will return the modified parameters with an operation code. The CA will print an error code returned by the TA regarding out-of-bounds issue.

Test Cases:

In-bounds:

```
# optee_example_hello_world
hello! ... Invoking TA to increment 42
TA incremented value to 43
```

Increment 42 to 43 is successful because 43 is within the bounds

```
# optee_example_hello_world
hello! ... Invoking TA to decrement 42
TA decremented value to 41
#
```

Decrement 42 to 41 because 41 is within the bounds

Out-of-bounds:

```
hello! ... Invoking TA to increment 42
optee_example_hello_world: Value is 142 which is out of bounds of [0, 100]
```

Incrementing 42 to 142 results in an out-of-bounds error because 142 is greater than 100.

```
hello! ... Invoking TA to decrement 42
optee_example_hello_world: Value is -58 which is out of bounds of [0, 100]
```

Decrementing 42 to -58 results in an out-of-bounds error -58 is less than 0.

Edge Cases:

```
# optee_example_hello_world
hello! ... Invoking TA to increment 100
optee_example_hello_world: Value is 101 which is out of bounds of [0, 100]
```

Incrementing 100 to 101 results in an out-of-bounds error because 101 is greater than 100.

```
# optee_example_hello_world
hello! ... Invoking TA to increment -1
TA incremented value to 0
#
```

Incrementing -1 to 0 results in successful execution because 0 is in the bounds.

```
# optee_example_hello_world
hello! ... Invoking TA to decrement 0
optee_example_hello_world: Value is -1 which is out of bounds of [0, 100]
```

Decrementing 0 to –1 results in an out-of-bounds error because -1 is less than 0.

```
# optee_example_hello_world
hello! ... Invoking TA to decrement 101
TA decremented value to 100
#
```

Decrementing 101 to 100 results in a successful execution because 100 is in the bounds.