

Test BC-HLR-20

" Ao concluir com sucesso, todo o processo de recebimento de uma imagem, o software deve liberar todos os buffers temporários e limpar variáveis de sessão"

Procedimento realizado

1. Deve-se copiar o sdkconfig (conforme [Configurações BC](#)) para o diretório `test` na raiz do projeto.
2. Deve-se abrir o ESP-IDF terminal com o caminho do projeto `.\modulo_bc\`
3. Utilizar o comando: `idf.py -C test build flash monitor`
4. Então após o build e flash bem sucedidos, irá se abrir o menu de escolha de testes unitários no terminal. Deve-se apertar Enter e posteriormente o 1 e enter para selecionar o primeiro teste

```
I (315) heap_init: At 4008E768 len 00011898 (70 KiB): IRAM
I (323) spi_flash: detected chip: generic
I (326) spi_flash: flash io: dio
I (331) main_task: Started on CPU0
I (341) main_task: Calling app_main()
Iniciando testes do componente auth e state_machine...

Press ENTER to see the list of tests.

Here's the test menu, pick your combo:
(1)    "auth_write_static_keys grava arquivos na particao keys" [auth]
(2)    "state_teardown_reset_globals limpa variaveis globais" [state_machine]

Enter test for running.
1
```

5. Deve-se verificar o resultado do teste

Código de teste

```

8 TEST_CASE("state_teardown_reset_globals limpa variaveis globais", "[state_machine]")
9 {
10     memset(&lur_file, 0xAA, sizeof(lur_file));
11     memset(hash, 0xBB, sizeof(hash));
12     memset(&req, 0xCC, sizeof(req));
13
14     req.opcode = 0xABCD;
15
16     static char dummy_name[] = "dummy";
17     filename = dummy_name;
18     opcode = 123;
19     n = 456;
20     upload_failure_count = 7;
21
22     auth_set_authenticated_for_test(true);
23     TEST_ASSERT_TRUE(auth_is_authenticated());
24
25     state_teardown_reset_globals();
26
27     lur_data_t expected_lur = {0};
28     tftp_packet_t expected_req = {0};
29     uint8_t expected_hash[sizeof(hash)] = {0};
30
31     TEST_ASSERT_EQUAL_MEMORY(&expected_lur, &lur_file, sizeof(lur_file));
32     TEST_ASSERT_EQUAL_UINT8_ARRAY(expected_hash, hash, sizeof(hash));
33     TEST_ASSERT_EQUAL_MEMORY(&expected_req, &req, sizeof(req));
34
35     TEST_ASSERT_NULL(filename);
36     TEST_ASSERT_EQUAL(0, opcode);
37     TEST_ASSERT_EQUAL_INT32(0, n);
38     TEST_ASSERT_EQUAL_UINT8(0, upload_failure_count);
39     TEST_ASSERT_FALSE(auth_is_authenticated());
40 }
41

```

Resultados obtidos

Log do teste realizado para verificar se as variáveis globais são liberados e zerados e log da função sendo chamada ao final de um upload

```

Here's the test menu, pick your combo:
(1) "auth_write_static_keys grava arquivos na particao keys" [auth]
(2) "state_teardown_reset_globals limpa variaveis globais" [state_machine]

Enter test for running.
2
Running state_teardown_reset_globals limpa variaveis globais...
I (389221) STATE_TEARDOWN: Limpando variáveis globais...
I (389221) auth: Resetando estado de autenticação
I (389241) STATE_TEARDOWN: Variáveis globais limpadas
D:/TCC_PES/gse_flfs/modulo_bc/components/state_machine/test/test_state_teardown.c:8:state_teardown_reset_globals limpa variaveis globais:PASS
Test ran in 37ms

-----
1 Tests 0 Failures 0 Ignored
OK
I (249696) tftp: Bloco 1998 recebido (512 bytes)
I (249876) tftp: Bloco 1999 recebido (512 bytes)
I (250096) tftp: Bloco 2000 recebido (512 bytes)
I (250276) tftp: Bloco 2001 recebido (72 bytes)
I (250296) tftp: Arquivo EMB-SN-007-137-045.bin recebido como temp.bin
I (250346) STATE_UPLOADING: RUNNING ST_UPLOADING
I (250346) STATE_UPLOADING: ACK enviado para hash (bloco 2002)
I (250346) STATE_UPLOADING: Endereço do cliente restaurado para IP=192...
I (250356) STATE_UPLOADING: EXIT ST_UPLOADING
I (250356) STATE_VERIFY: INIT ST_VERIFY
I (250406) STATE_VERIFY: RUNNING ST_VERIFY
I (250406) STATE_VERIFY: Hash SHA-256 conferido com sucesso.
I (250406) STATE_VERIFY: EXIT ST_VERIFY
I (250406) STATE_SAVE: INIT ST_SAVE
I (250456) STATE_SAVE: RUNNING ST_SAVE
I (250456) storage: Finalizing firmware file: temp.bin -> final.bin
I (256576) storage: Removed existing final.bin
I (256726) storage: Successfully renamed temp.bin to final.bin
I (256726) STATE_SAVE: EXIT ST_SAVE
I (256726) STATE_TEARDOWN: INIT ST_TEARDOWN
I (256776) STATE_TEARDOWN: RUNNING ST_TEARDOWN
I (256776) arinc: LUS initialized: status=<003, counter=2, ratio=100
I (256776) tftp: Iniciando WRQ para envio de FINAL_LOAD.LUS
I (256816) tftp: Cliente mudou para TID (porta) 49783
I (256826) tftp: Arquivo LUS enviado com sucesso
I (256826) STATE_TEARDOWN: Limpando variáveis globais...
I (256826) auth: Resetando estado de autenticação
I (256826) STATE_TEARDOWN: Variáveis globais limpadas
I (256836) STATE_TEARDOWN: EXIT ST_TEARDOWN
I (256836) STATE_MAINT_WAIT: INIT ST_MAINT_WAIT
I (256846) STATE_MAINT_WAIT: Sistema já inicializado, pulando configuração WiFi/socket
I (256856) auth: Carregando chaves da partição
I (256856) auth: Chaves carregadas com sucesso

```

Imagem: Nenhuma imagem selecionada

Logs:

```
[TFTP-ARINC] HASH enviado e ACK recebido.
[TFTP-ARINC] Socket de transferência (BIN) fechado
[ARINC] BIN e HASH servidos com sucesso.
[ARINC] PASSO 5/5: Aguardando LUS 100%...
[TFTP-ARINC] Aguardando WRQ (LUS) no socket principal...
[TFTP-ARINC] WRQ para 'FINAL_LOAD.LUS' do módulo.
[ARINC] LUS 100% recebido.
=====
[ARINC] Fluxo de upload concluído com sucesso.
=====
(ok) Transferência concluída.
[TFTP-OK] Socket principal fechado
[WORKER] Thread encerrada e sockets limpos.
```

Progresso:

Transferir
Selecionar Imagem
Sair