

LFR-FSW - Task #4026

Analyse GCOV

13/02/2023 11:10 AM - Veronique bouzid

| | | | |
|------------------------|----------------|------------------------|------------|
| Status: | Closed | Start date: | 13/02/2023 |
| Priority: | Urgent | Due date: | |
| Assignee: | Alexis Jeandet | % Done: | 0% |
| Category: | | Estimated time: | 0.00 hour |
| Target version: | 3.3 | Spent time: | 0.00 hour |
| revision: | r0 | | |

Description

Début de la couverture de Test en 3.3.0.16

Une quinzaine de tests ont été joués. Avant de pousser plus loin, il faudrait traiter ses tests et évaluer la couverture atteinte.

Sur pc-solar3, dans le répertoire /opt/VALIDATION_R3.3/GCOV/3.3.0.16, voici les fichiers à traiter:

```
gcov_out_2023-02-13 09:10:52.635355.txt
gcov_out_2023-02-13 09:12:32.358845.txt
gcov_out_2023-02-13 09:15:53.533775.txt
gcov_out_2023-02-13 09:17:58.219268.txt
gcov_out_2023-02-13 09:23:23.893858.txt
gcov_out_2023-02-13 09:33:27.972367.txt
gcov_out_2023-02-13 09:37:47.458757.txt
gcov_out_2023-02-13 09:39:33.827127.txt
gcov_out_2023-02-13 09:41:52.977440.txt
gcov_out_2023-02-13 10:23:51.069120.txt (SVS-0002 à SVS-0010 + SVS-0059)
```

History

#1 - 14/02/2023 04:32 PM - Alexis Jeandet

Pour le moment, je n'arrive pas à avoir la couverture, j'ai tout à 0%.

#2 - 14/02/2023 04:47 PM - Alexis Jeandet

- File fsw-gcov-test added

Est-ce que tu peux faire un test avec ce fsw? J'ai un doute sur la procédure.

#3 - 15/02/2023 08:06 AM - Veronique bouzid

Sur pc-solar3

Installation de la dernière version du fsw dans /opt/LFR/LFR-FSW/3.3.0.16.

Lancement du test /opt/VALIDATION_R3.3/lfrverif/LFR_SVS/SVS-0003/loop_tm_lfr_tc_exe.py

Les 2 fichiers résultats sont dans /opt/VALIDATION_R3.3/GCOV

~~rw-rw-r~~. 1 validation validation 21098 15 févr. 07:54 console_2023-02-15 07:49:25.729931.txt

~~rw-rw-r~~. 1 validation validation 49778 15 févr. 07:54 gcov_out_2023-02-15 07:49:25.729748.txt

Le script qui initialise la liaison serie est **record_lfr_console.py** (script que tu avais écrit pour les précédentes versions).

#4 - 16/02/2023 11:53 AM - Alexis Jeandet

- File analysis.zip added

Voici l'analyse de couverture totale et pour chaque test.

#5 - 20/02/2023 10:47 AM - Veronique bouzid

- Status changed from New to In Progress

Bonjour,

J'ai analysé la couverture pour identifier les parties de code non jouées.

Sur pc-solar3, tu trouvera dans /opt/VALIDATION_R3.3/GCOV/3.3.0.16

7 tests supplémentaires.

[validation@pc-solar3 3.3.0.16]\$ ls 2023-02-20

```
console_2023-02-20 07:55:42.383724.txt console_2023-02-20 08:39:40.153165.txt gcov_out_2023-02-20 08:04:00.945897.txt
console_2023-02-20 07:59:33.397393.txt console_2023-02-20 10:20:08.631000.txt gcov_out_2023-02-20 08:36:13.910020.txt
console_2023-02-20 08:02:21.477948.txt gcov_out_2023-02-20 07:55:42.370695.txt gcov_out_2023-02-20 08:39:40.152985.txt
console_2023-02-20 08:04:00.946080.txt gcov_out_2023-02-20 07:59:33.397202.txt gcov_out_2023-02-20 10:20:08.630817.txt
console_2023-02-20 08:36:13.910209.txt gcov_out_2023-02-20 08:02:21.477746.txt
```

Peux-tu donc générer la couverture de chaque test et refaire du coup la couverture totale en prenant en compte l'ensemble des tests joués le 13/2/2023 et le 20/2/2023.

#6 - 20/02/2023 11:49 AM - Alexis Jeandet

- File analysis.zip added

Voici le résultat, j'ai réussi à enlever les PRINTF mais pas les fichiers gcov.

#7 - 28/03/2023 02:02 PM - Veronique bouzid

- File doc06963020230220112648.pdf added

- Status changed from In Progress to Closed

Explication concernant la couverture dans ce fichier

*file:///C:/Users/bouzid/LFR/analysis/analysis/total_coverage/src/tc_tm/tc_acceptance.c.gcov.html

Cette fonction n a pas été appelée mais ces lignes de codes ont été jouées.*

```
0 : void GetCRCAsTwoBytes(
104 :     const unsigned char* data, unsigned char* crcAsTwoBytes, unsigned int sizeOfData)
105 :     {
106 :     /** This function calculates a two bytes Cyclic Redundancy Code.
107 :     *
108 :     * @param data points to a buffer containing the data on which to compute the CRC.
109 :     * @param crcAsTwoBytes points to a two bytes buffer in which the CRC is stored.
110 :     * @param sizeOfData is the number of bytes of data used to compute the CRC.
111 :     *
112 :     * The specification of the Cyclic Redundancy Code is described in the following document:
113 :     * ECSS-E-70-41-A.
114 :     *
115 :     */
116 :
117     5775 :     unsigned int Chk = CRC_RESET;
118     176635 :     for (unsigned int j = 0; j < sizeOfData; j++)
119 :     {
120     341720 :         Chk = Crc_opt(data[j], Chk);
121 :     }
122     5775 :     crcAsTwoBytes[0] = (unsigned char) (Chk >> SHIFT_1_BYTE);
123     5775 :     crcAsTwoBytes[1] = (unsigned char) (Chk & BYTE1_MASK);
```

Réponse d'Alexis

En fait c'est normal, je pense, il faut bien regarder toutes les lignes. Ce qui n'est pas exécuté c'est le code de associé au call de la fonction car elle est inliné, le reste de la fonction est bien appelé et le bon nombre de fois, 5775 fois pour la ligne 117 ce qui correspond au nombre de fois qu'on est passé à la ligne 514.

Voici la fonction tc_check_crc désassemblée, tu peux voir que GetCRCAsTwoBytes est bien inlinée:

```

498
499 int tc_check_crc(const ccsdsTelecommandPacket_t* const TCPacket, unsigned int length,
500 unsigned char* computed_CRC)
501 {
0x4000fcfc <+0>: popf
0x4000fcfd <+1>: jecxz 0x4000fcbe <tc_check_length+1106>
0x4000fcff <+3>: mov 0x3f000005,%al

502 /** This function checks the CRC validity of the corresponding TeleCommand packet.
503
504 * @param TCPacket points to the TeleCommand packet to check.
505 * @param length is the length of the TC packet.
506 *
507 * @return Status code CCSDS_TM_VALID or INCOR_CHECKSUM.
508 *
509 */
510
511 int status = CCSDS_TM_VALID;
512 const unsigned char const CCSDSContent = (const unsigned char*)TCPacket->packetID;
0x4000fd74 <+120>: mov %al,(%esi)
0x4000fd76 <+122>: add %al,(%edx,%eax,8)
0x4000fd79 <+125>: or %esp,(%eax)
0x4000fd7b <+127>: add -0x7f001ff8(%esi),%eax
0x4000fd81 <+133>: mov 0x100301c0,%al
0x4000fd86 <+138>: add $0x80027d,%eax
0x4000fd8b <+143>: or %al,-0x3b979ff0(%edx)
0x4000fd91 <+149>: sbb %ah,0x18(%edx)
0x4000fd94 <+152>: xchg %al,0x408401e0(%eax)
0x4000fd9a <+158>: mov 0x6238c400,%al
0x4000fd9f <+163>: sbb %al,-0x6ef71f39(%ecx)
0x4000fda5 <+169>: call 0x58d8ffca
0x400fdaa <+174>: bound %edx,(%eax)
0x4000fdac <+176>: lcall $0x2043,$0x98016083
0x4000fdb3 <+183>: add %bl,%al
0x4000fdb5 <+185>: cmp %ah,0x10(%edx)

513
514 GetCRCAsTwoBytes(CCSDSContent, computed_CRC,
0x4000fd00 <+4>: add $0x1b3f0000,%eax
0x4000fd05 <+9>: adc %al,0xa310847d
0x4000fd0b <+15>: call *(%edi)
0x4000fd10 <+20>: lcall $0xe012,$0x96686013
0x4000fd18 <+28>: mov %al,(%esi)
0x4000fd1a <+30>: pusha
0x4000fd1b <+31>: add $0x201082,%eax
0x4000fd28 <+44>: rcrb (%ebx)
0x4000fd2a <+46>: bound %ecx,(%eax)
0x4000fd30 <+52>: xchg %eax,%edx
0x4000fd31 <+53>: andb $0x90,0x1(%eax)
0x4000fd35 <+57>: inc %edx
0x4000fd36 <+58>: and %al,(%eax)
0x4000fd38 <+60>: sarb (%ebx)
0x4000fd3a <+62>: bound %ecx,(%eax)

515 length + CCSDS_TC_TM_PACKET_OFFSET
516 - BYTES_PER_CRC); // 2 CRC bytes removed from the calculation of the CRC
517
518 if (computed_CRC0 != CCSDSContent[length + CCSDS_TC_TM_PACKET_OFFSET - BYTES_PER_CRC])
519 {
520 status = INCOR_CHECKSUM;
521 }
522 else if (computed_CRC1 != CCSDSContent[length + CCSDS_TC_TM_PACKET_OFFSET - 1])
0x4000fdb8 <+188>: mov $0x6,%dl
0x4000fdb9 <+190>: add %bl,(%ecx)
0x4000fdbc <+192>: test %cl,(%eax)
0x4000fdbe <+194>: mov 0x600ec6ff,%al

```

```
0x4000fdc3 <+199>: or %eax,0x20380a0(%eax)
0x4000fdc9 <+205>: addb $0x7,(%eax)
0x4000fdcc <+208>: mov $0x10,%al
0x4000fdce <+210>: and %al,(%edi)
0x4000fdd0 <+212>: les (%eax),%ebx
0x4000fdd2 <+214>: bound %esp,(%eax)
0x4000fdd4 <+216>: mov $0x10,%al
0x4000fdd6 <+218>: and %al,(%edx)
0x4000fdd8 <+220>: xchg %al,0x408401e0(%eax)
0x4000fdde <+226>: mov 0x6238c400,%al
0x4000fde3 <+231>: and %al,-0x7ef71f39(%ecx)
```

```
523 {
524 status = INCOR_CHECKSUM;
525 }
526
527 return status;
528 }
0x4000fde4 <+232>: add $0xe88108e0,%edi
0x4000fdea <+238>: add %al,(%eax)
```

Cette couverture est celle livrée dans le datapack.

Cela peut être amélioré, par exemple jouer les tests du SVS-0018 pour valider DESTINATION_ID. Voir également les TC_LFR_UPDATE_XXX.

Files

| | | | |
|-----------------------------|---------|------------|------------------|
| fsw-gcov-test | 4.41 MB | 14/02/2023 | Alexis Jeandet |
| analysis.zip | 6.08 MB | 16/02/2023 | Alexis Jeandet |
| analysis.zip | 9.6 MB | 20/02/2023 | Alexis Jeandet |
| doc06963020230220112648.pdf | 273 KB | 28/03/2023 | Veronique bouzid |