

←

RTEMS GSOC 2017 Project: Beaglebone BSP Projects



My chinese Blog is more detail: <http://blog.csdn.net/hahachenchen789> I am porting my chinese Blog to Google Blogger here.

Enable CFDP protocol transmit on ION-DTN



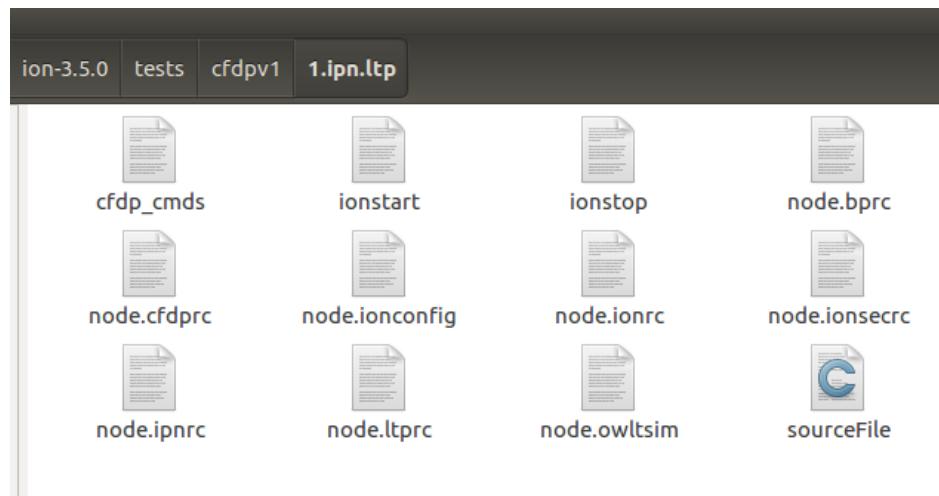
七月 06, 2017

CFDP (CCSDS File Delivery Protocol), another application-layer service that is not part of the DTN architecture but utilizes underlying DTN protocols. CFDP performs the segmentation, transmission, reception, reassembly, and delivery of files in a delay-tolerant manner. ION's implementation of CFDP conforms to the "class 1" definition of the protocol in the CFDP standard, utilizing DTN (BP, nominally over LTP) as its "unitdata transport" layer.

That is to say, CFDP is another application layer protocol based on the DTN protocol stack, which is mainly used for file transmission.

So, how to open the protocol transport in ION?

ION gives a test of the demo CFDP, located in the tests/cfdpv1 folder, you can see contains two nodes, that is the point to point to each other, first look at node 1, contains a lot of RC files and some startup files, as shown below:



Among them, we are most concerned about is the ionstart file, because we all know that the DTN protocol is open using the ionstart script command.

The ionstart script reads as follows;

```

01. # shell script to get node running
02. #!/bin/bash
03. ionadmin node.ionrc
04. sleep 1
05. ionadmin ..../global.ionrc
06. sleep 1
07. ionsecadmin node.ionsecrc
08. sleep 1
09. owltsim node.owltsim -v &
10. sleep 1
11. ltpradmin node.ltprc
12. sleep 1
13. bpradmin node.bprc
14. sleep 1
15. cfdpadmin node.cfdprc

```

← RTEMS GSOC 2017 Project: Beaglebone BSP Projects



```
01. ./ionstart
```

If you find some permission errors when running, you should add sudo before the command

In this way, the script runs the individual RC files, noting that in the first configuration file, the node.ionrc file:

```
01. 1 1 node.ionconfig
02. s
03. m horizon +0
```

We use the node.ionconfig file to configure

```
01. wmKey 66136
02. sdrName ion1
03. wmSize 1000000
04. configFlags 1
05. heapWords 1000000
```

This configuration file cannot be used properly. Size is too small. It should be changed to the following configuration:

```
01. wmSize 500000000
02. wmAddress 0
03. #configFlags = 14 means the ION data store is in a file (in /usr/ion)
04. #only, transactions are reversible, and object boundaries are enforced.
05. configFlags 1
06. heapWords 150000000
07. pathName /usr/io
```

In this way the ionstart works properly.

After opening the CFDP service, you should enter the command to control the CFDP transport file. Here you can use the following command:

```
01. cfdptest cfdp_cmds
```

Where "cfdptest" is the command, followed by the "cfdp_cmds" is the command script, the file is also in the directory of node 1, and then explain the command script in detail:

```
01. z 2
02. d 2
03. f sourceFile
04. t receivedFile
05. a 10
06. &
07. z 10
08. d 2
09. f sourceFile
10. t receivedFile
11. a 10
12. l
13. z 18
14. q
```

z [number of seconds to pause]

The pause command. When cfdptest is running in interactive mode, this command causes the console input processing thread to pause the indicated number of seconds (defaulting to 1) before processing the next command. This is provided for use in test scripting.

← RTEMS GSOC 2017 Project: Beaglebone BSP Projects



d <destination CFDP entity ID number>

The destination command. This command establishes the CFDP entity to which the next file transmission operation will be directed. CFDP entity numbers in ION are, by convention, the same as BP node numbers.

The "D" command is the destination node.

f <source file path name>

The from command. This command identifies the file that will be transmitted when the next file transmission operation is commanded.

"F" is easy to understand, the path name and file name of the source file that is transmitted

t <destination file path name>

The to command. This command provides the name for the file that will be created at the receiving entity when the next file transmission operation is commanded.

"T" is the destination path name for transmitting the past, and it is also well understood.

a <latency in seconds>

The closure latency command. This command establishes the transaction closure latency for all subsequent file transmission operations. When it is set to zero, the file transmission is "open loop" and the CFDP transaction at the sending entity finishes when the EOF is sent. Otherwise, the receiving CFDP entity is being asked to send a "Finished" PDU back to the sending CFDP entity when the transaction finishes at the receiving entity. Normally the transaction finishes at the sending entity only when that Finished PDU is received. However, when closure latency seconds elapse following transmission of the EOF PDU prior to receipt of the Finished PDU, the transaction finishes immediately with a Check Timer fault.

"a" shutdown waiting command. This command is to wait for x seconds after the process is closed, that is, CFDP transmission, followed by the parameter is x seconds, and if set to 0, then it will cycle.

& The send command. This command initiates file transmission as configured by the most recent preceding d, f, t, and a commands.

"&" Send command.

| The get command. This command causes a request for file transmission to the local node, subject to the parameters provided by the most recent preceding f, t, and a commands, to be sent to the entity identified by the most recent preceding d command.

"!" command to get results.

Therefore, when you parse and run the cfdfp_cmds script file by using cfdfptest, we can get the generated received file under the node 2 directory of the cfdfpv1 of the destination host.

Done.

← RTEMS GSOC 2017 Project: Beaglebone BSP Projects



输入您的评论...

此博客中的热门博文

Developed the I2C driver for Beaglebone Black BSP

五月 22, 2017



My first part of GSOC project is finish the I2C driver left by GSOC 2016 student Punit Vara.
I reference the U-Boot and FreeBSD i2c code for develop.

[阅读全文](#)

...

Port WPA supplicant to RTEMS-libbsd

八月 11, 2017

Last blog post, i write about how to import wpa from FreeBSD.
This blog post, i will introduce how to port WPA to RTEMS-libbsd.

[阅读全文](#)

...

Add USB dongle rtl8188eu support for RTEMS-libbsd on BBB BSP

八月 08, 2017

After add the USB support for RTEMS-libbsd on Beaglebone Black BSP. The next step is to add the USB dongle driver support, i choose the rtl8188eu chip as my USB WiFi dongle.

[阅读全文](#)

...

由 Blogger 提供支持

主题背景图片创建者 : Michael Elkan



UNKNOWN

[← RTEMS GSOC 2017 Project: Beaglebone BSP Projects](#)

归档



举报滥用情况