Activity: Reduce cFS Resource Requirements

This activity will Inspect the memory footprint of cFS and explore configuration options to reduce it.

# Analyze the Resource Requirements

|  |  |  |
| --- | --- | --- |
| 1 | Rebuild cFS. | debian@beaglebone:~/cFS$ make distclean  rm -rf "build"  debian@beaglebone:~/cFS$ make prep  mkdir -p "build"  <snip>  echo " -DCMAKE\_INSTALL\_PREFIX=/exe -DCMAKE\_BUILD\_TYPE=debug" > "build/.prep"  debian@beaglebone:~/cFS$ make  make --no-print-directory -C "build" mission-all  <snip>  Built target mission-all  debian@beaglebone:~/cFS$ make install  make --no-print-directory -C "build" DESTDIR="/home/debian/cFS/build" mission-install  <snip>  Built target mission-install |
| 2 | Observe the segment sizes of the executable. In particular, note the size of bss, the uninitialized data segment. | debian@beaglebone:~/cFS$ cd build/exe/cpu1/  debian@beaglebone:~/cFS/build/exe/cpu1$ size core-cpu1  text data bss dec hex filename  259672 8164 1333504 1601340 186f3c core-cpu1 |
| 3 | To get more information, generate a memory map file. Ideally, we could use cmake’s add\_link\_options, but it is not available in the version of cmake on the BeagleBone AI. Instead, we’ll manually modify the link command. Do you see how the memory is being used? It can take a lot of analysis! | debian@beaglebone:~/cFS$ cd build/cpu1/cpu1/CMakeFiles/core-cpu1.dir/  debian@beaglebone:~/cFS/build/cpu1/cpu1/CMakeFiles/core-cpu1.dir$ cat link.txt  /usr/bin/gcc -Wall -D\_XOPEN\_SOURCE=600 -g -Wl,--export-dynamic CMakeFiles/core-cpu1.dir/src/target\_config.c.o -o core-cpu1 -rdynamic -Wl,--whole-archive ../cfe\_core\_default\_cpu1/libcfe\_core\_default\_cpu1.a ../psp/pc-linux/libpsp-pc-linux.a ../osal/libosal.a -Wl,--no-whole-archive -lpthread -ldl -lrt -Xlinker -Map=output.map  debian@beaglebone:~/cFS/build/cpu1/cpu1/CMakeFiles/core-cpu1.dir$ cd ../../../../exe/cpu1/  debian@beaglebone:~/cFS/build/exe/cpu1$ rm core-cpu1  debian@beaglebone:~/cFS/build/exe/cpu1$ cd ../../..  debian@beaglebone:~/cFS$ make install  make --no-print-directory -C "build" DESTDIR="/home/debian/cFS/build" mission-install  <snip>  Built target mission-install  debian@beaglebone:~/cFS$ cd build/cpu1/cpu1  debian@beaglebone:~/cFS/build/cpu1/cpu1$ cat output.map  Archive member included to satisfy reference by file (symbol)  ../cfe\_core\_default\_cpu1/libcfe\_core\_default\_cpu1.a(cfe\_es\_api.c.o)  (--whole-archive)  ../cfe\_core\_default\_cpu1/libcfe\_core\_default\_cpu1.a(cfe\_es\_apps.c.o)  (--whole-archive)  ../cfe\_core\_default\_cpu1/libcfe\_core\_default\_cpu1.a(cfe\_es\_cds.c.o)  (--whole-archive)  ../cfe\_core\_default\_cpu1/libcfe\_core\_default\_cpu1.a(cfe\_es\_cds\_mempool.c.o)  (--whole-archive)  <snip>  .gnu.attributes  \*(.gnu.attributes)  .note.gnu.arm.ident  \*(.note.gnu.arm.ident)  /DISCARD/  \*(.note.GNU-stack)  \*(.gnu\_debuglink)  \*(.gnu.lto\_\*)  OUTPUT(core-cpu1 elf32-littlearm) |
| 4 | Observe a large amount of the bss segment is used by CFE\_TBL\_TaskData and CFE\_SB. | CFE\_TBL\_TaskData 0x8ce88 ../cfe\_core\_default\_cpu1/libcfe\_core\_default\_cpu1.a(cfe\_tbl\_task.c.o)  CFE\_SB 0x862e0 ../cfe\_core\_default\_cpu1/libcfe\_core\_default\_cpu1.a(cfe\_sb\_task.c.o) |
| 5 | Can you reduce the memory footprint of the executable? Make 3 changes to reduce the total memory used by text, data, and bss to less than 1048576. How do you know these changes are suitable? Be sure you know how the settings impact the operation of the application. You may have to research the format of the linker output file. SB’s Send Statistics Pkt provides hints. | Consider the following as two options to reduce memory. Where will you find a third change where you save substantial memory?  debian@beaglebone:~/cFS$ git diff sample\_defs/cpu1\_platform\_cfg.h  diff --git a/sample\_defs/cpu1\_platform\_cfg.h b/sample\_defs/cpu1\_platform\_cfg.h  index be21bc8..a1b1a26 100644  --- a/sample\_defs/cpu1\_platform\_cfg.h  +++ b/sample\_defs/cpu1\_platform\_cfg.h  @@ -139,7 +139,7 @@  \*\* This parameter has a lower limit of 512 and an upper limit of UINT\_MAX (4 Gigabytes).  \*\*  \*/  -#define CFE\_PLATFORM\_SB\_BUF\_MEMORY\_BYTES 524288  +#define CFE\_PLATFORM\_SB\_BUF\_MEMORY\_BYTES 245760  /\*\*  @@ -1621,7 +1621,7 @@  \*\* \par Limits  \*\* The cFE does not place a limit on the size of this parameter.  \*/  -#define CFE\_PLATFORM\_TBL\_BUF\_MEMORY\_BYTES 524288  +#define CFE\_PLATFORM\_TBL\_BUF\_MEMORY\_BYTES 245760  /\*\*  \*\* \cfetblcfg Maximum Size Allowed for a Double Buffered Table |