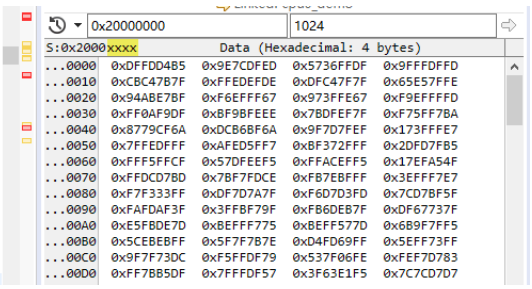


# 常用newlib库函数测试

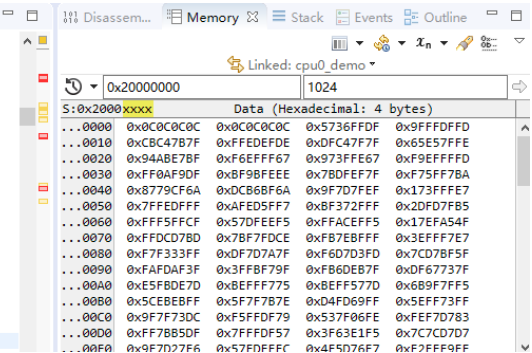
由于newlib库函数数量庞大，无法一一测试验证，这里提取几个裸机中常用的库函数进行测试验证，这几个函数分别是memset、memcpy、memcmp、strlen。

## memset测试

```
37
38 int main (void)
39 {
40
41     #if 1
42     u32 *p = 0x20000000;
43     memset(p,0xc,8);
44     asm __volatile__ ("nop");
45     #endif
46
47
48
49
50
51
52
53
54
```



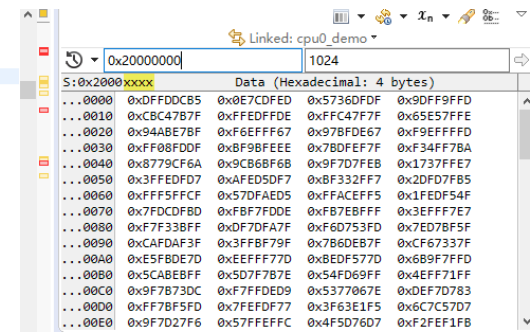
```
135
136 }
137
138 int main (void)
139 {
140
141     #if 1
142     u32 *p = 0x20000000;
143     memset(p,0xc,8);
144     asm __volatile__ ("nop");
145     #endif
146
147
148
149
150
151
152
153
154
```

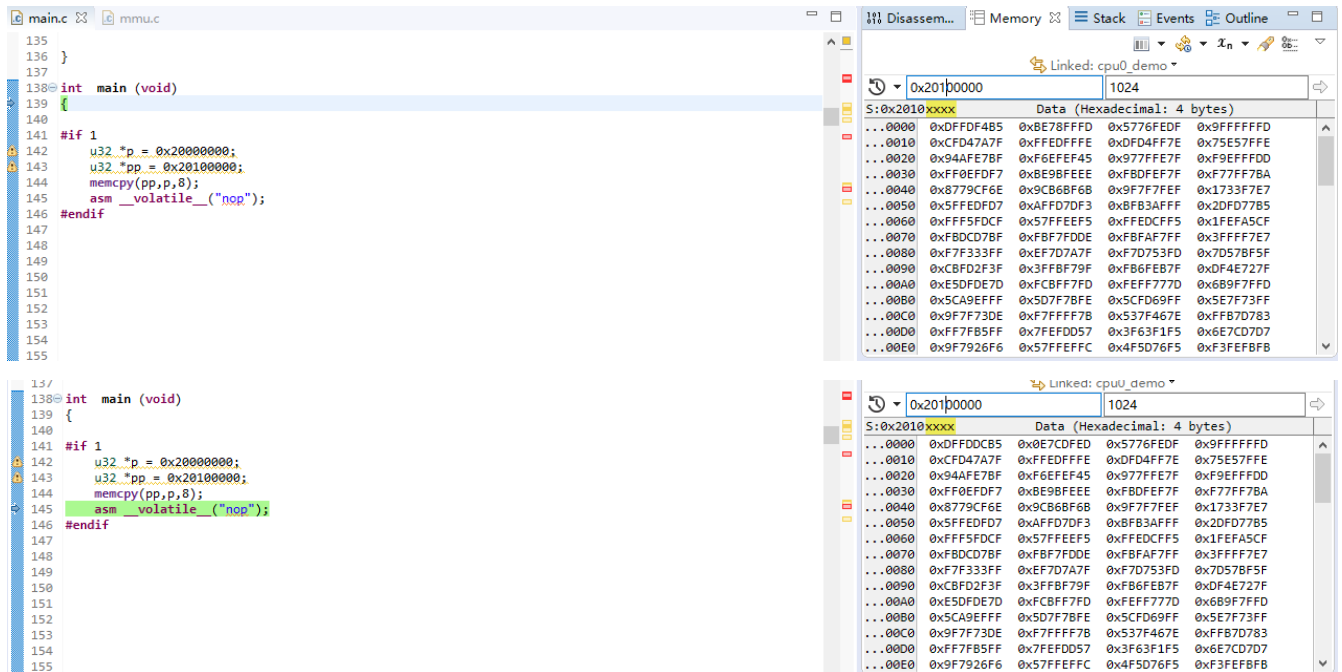


上图展示的是通过memset将地址0x20000000处开始的8字节设置成0x0c，从第2图的内存显式中可以看到确实该处8字节都变成了0x0c。因此memset测试没有问题。

## memcpy测试

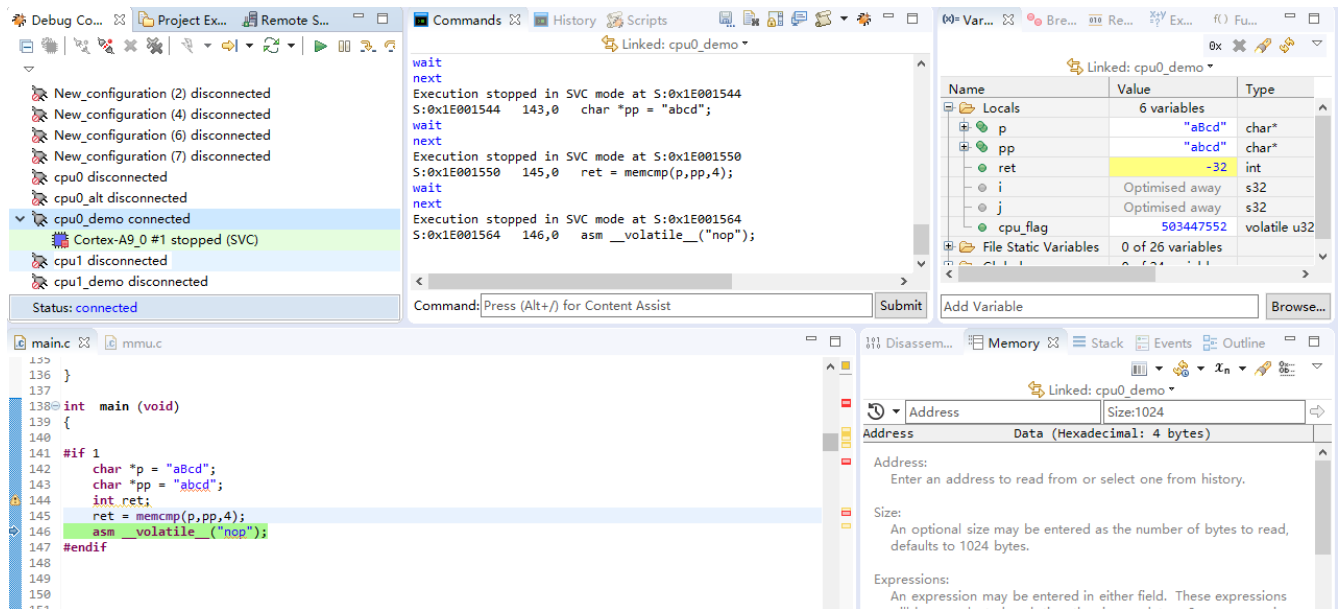
```
135
136 }
137
138 int main (void)
139 {
140
141     #if 1
142     u32 *p = 0x20000000;
143     u32 *pp = 0x20100000;
144     memcpy(pp,p,8);
145     asm __volatile__ ("nop");
146     #endif
147
148
149
150
151
152
153
154
```

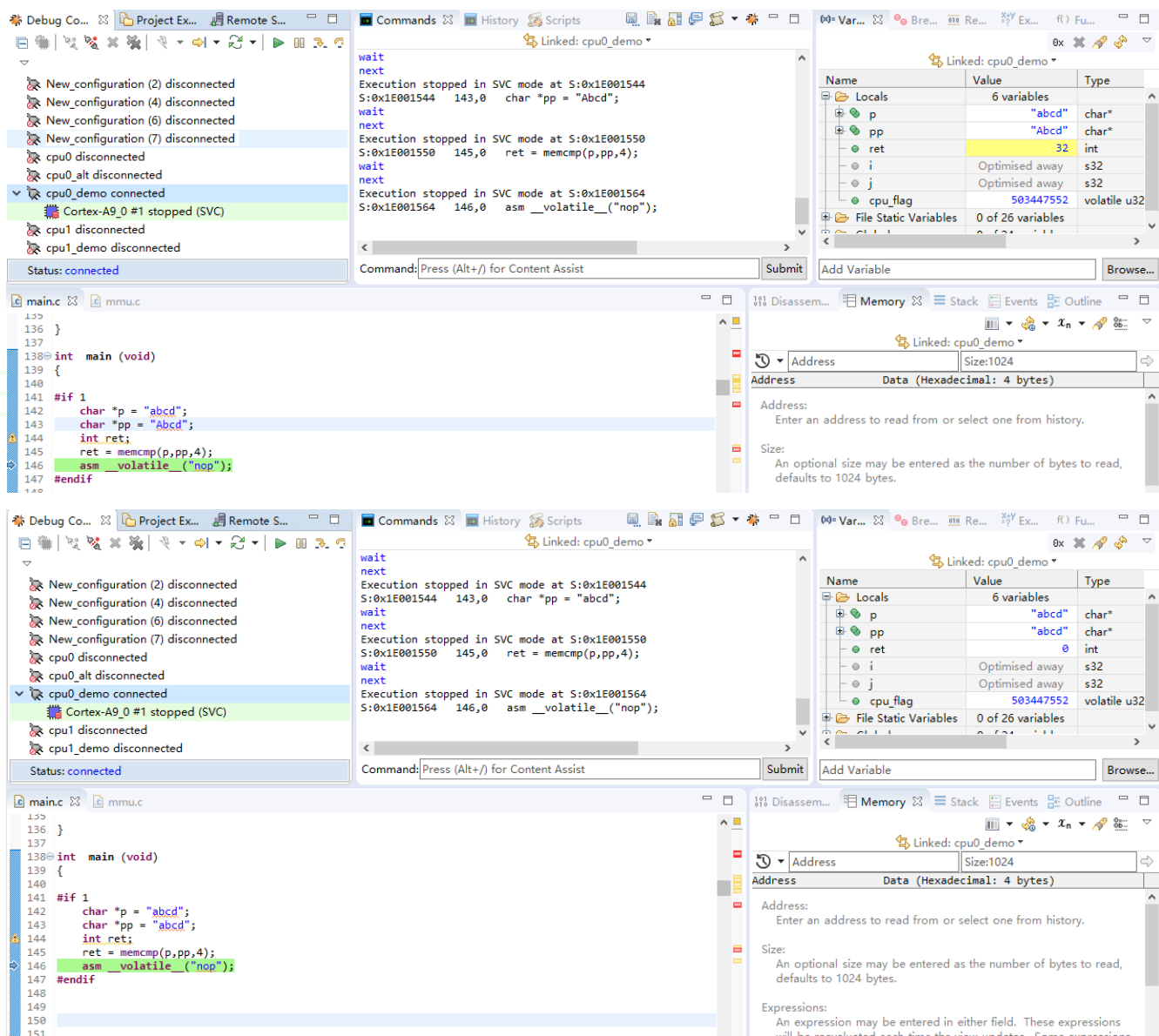




上图展示的是用memcpy将p指向的0x20000000处的8字节内容复制到pp指向的0x20100000处。从图中可以看到p处的8字节内容为0xDFFDDCB5,0xE7CDFED。未复制前pp处的8字节内容为0xDFFDF4B5,0xBE78FFFD。复制后pp处的8字节内容变为0xDFFDDCB5,0xE7CDFED，和p处的内容一样。因此memcpy测试没有问题。

## memcpy测试





上图分别将p和pp指向的字符串用memcmp进行比较，第一图返回值为-32,因为aBcd中的B比abcd中的b小32，第二图返回值为32，因为abcd中的a比Abcd中的A大32，第三图返回值为0，因为比较的字符串一样。因此memcmp测试没有问题。

## strlen测试

The screenshot displays a debugger interface with the following components:

- Left Panel (Device Tree):** Shows a list of components including 'New\_configuration (2) disconnected', 'New\_configuration (4) disconnected', 'New\_configuration (6) disconnected', 'New\_configuration (7) disconnected', 'cpu0 disconnected', 'cpu0\_alt disconnected', 'cpu0\_demo connected', 'Cortex-A9\_0 #1 stopped (SVC)', 'cpu1 disconnected', and 'cpu1\_demo disconnected'. The status is 'connected'.
- Command Window:** Shows the execution flow with commands like 'wait', 'next', and 'Execution stopped in SVC mode at S:0x1E001538'. It also shows the return value of 'strlen(p)' as 144,0.
- Source Code Window:** Displays the C code in 'main.c' and 'mmu.c'. The code defines a string 'p' as 'abcdefg', calculates its length using 'strlen', and prints it. The line 'ret = strlen(p);' is highlighted.
- Locals Window:** Shows the local variables for the current function. The variable 'p' is of type 'char\*' and contains the value 'abcdefg'. The variable 'ret' is of type 'size\_t' and contains the value 7.
- Memory Window:** Shows the memory address and data for the current function. The address is '0x1E001538' and the data is 'Data (Hexadecimal: 4 bytes)'.

上图展示的是用strlen计算一个字符串“abcdefg”的长度，从结果看到返回值为7,刚好为该字符串的长度。因此strlen测试没有问题。