How to debug you GBA codes?

Computer Organization and Assembly 2010

Tz-Huan Huang

Debugging tools

- A debugger
  - gdb, insight, etc
- An emulator which supports debugging with your debugger
  - Visual boy advance (VBA), etc
- Unfortunately, the win32 version of VBA doesn’t support gdb debugging any more…

Visual Boy Advance, SDL version

- Fortunately, VBA-SDL bundles a tiny debugger itself.
- Go to http://vba.ngemu.com to download it.

Basic steps

1. Build your assignment
2. Find the address of “myfilter’’
3. Run VBA-SDL with debugging mode
4. Break your code at the address of “myfilter’’
5. debug your codes
6. Repeat 1-5 until no bug left, then you can submit your assignment

1. Build your program

- (Install devkitPro first.)
- Launch the command-line box
- Go to the directory where you put the hw3 template.
  - For example, if you unzip hw3.zip to c:\asm2010\hw3, then type the following command to go there:
  - `c:\> cd c:\asm2010\hw3`
- Type ``**make**’’ to build your program.

2. Find the address of “myfilter”

- Use the “nm” tool bundled in devkitPro.
- For example, if you install devkitPro int c:\devkitPro, you will find (either is ok)
  - `c:\devkitPro\devkitARM\bin\arm-eabi-nm.exe`, or
  - `c:\devkitPro\devkitARM\arm-eabi\bin\nm.exe`
- Run “nm hw3.elf” to find the address of “myfilter”

```
C:\> c:\...\arm-eabi-nm.exe hw3.elf
  | grep myfilter
[ADDR] T myfilter
```

3. Run VBA-SDL with debugging

- Download VBA-SDL and unzip it
- In command-line box, type
  - C:\> **[path to VBA-SDL]\VisualBoyAdvance-SDL –d hw3.gba**

4. Break your code at "myfilter"

- In VBA-SDL debugger, add a breakpoint:
  - **ba [addr]**

  where [addr] is the address of "myfilter"
- Type **c** to continue the program until the break point

5. Debug your codes

- Useful commands:

– ba	Adds an ARM breakpoint
– bd	Deletes a breakpoint
– bl	Lists breakpoints
– c	Continues execution
– d	Disassembles instructions
– eb	Modify memory location (byte)
– eh	Modify memory location (half-word)
– ew	Modify memory location (word)
– mb	Shows memory contents (bytes)
– mh	Shows memory contents (half-words)
– mw	Shows memory contents (words)
– n	Executes the next instruction
– q	Quits the emulator
– r	Shows ARM registers

- Use **h** or **?** for more information

Happy debugging!