SCC150 – MIPS/Assembly Week 14/16 Assessed Practical

Angie Chandler
SCC – Lancaster University
Week 14/16

Assessed Exercise

- This exercise is assessed
- You should work alone
- The deadline is Friday Week 16, 4pm GMT
- You must submit your work before the deadline
- You should submit
  - ASM file containing your code
  - Text file (doc, pdf, txt, odt) containing your predicted marks (very short!)

Be careful

- Do not share code or code solution in teams chat or the forum.
  - If you need help, talk to one of the TAs one-to-one.
  - All code submissions are checked by plagiarism checking tools.

- If you have any questions about coursework please ask academics.
  - Do not trust information that you read on group chat.
  - Exploit lab time to discuss any problem with TAs and academics.
  - **If you are not sure, ask us.**

- If you are facing any difficulties and you cannot meet deadlines, please contact our teaching office as soon as possible.
  - Teaching academics cannot grant extensions.

Coursework context

- Build a 2D drawing application using MARS.
  - Support a simple cli (command line interface).
- Supported operations:
  - Cls: paint all pixels to a specific colour
  - Stave: paint black all pixels of 5 equally spaced horizontal lines.
  - Note: play a musical note (syscall) and draw a square in the correct place
- Your solution should follow the coursework requirement to get full marks (e.g. use procedure where asked, follow procedure rules).

Outline

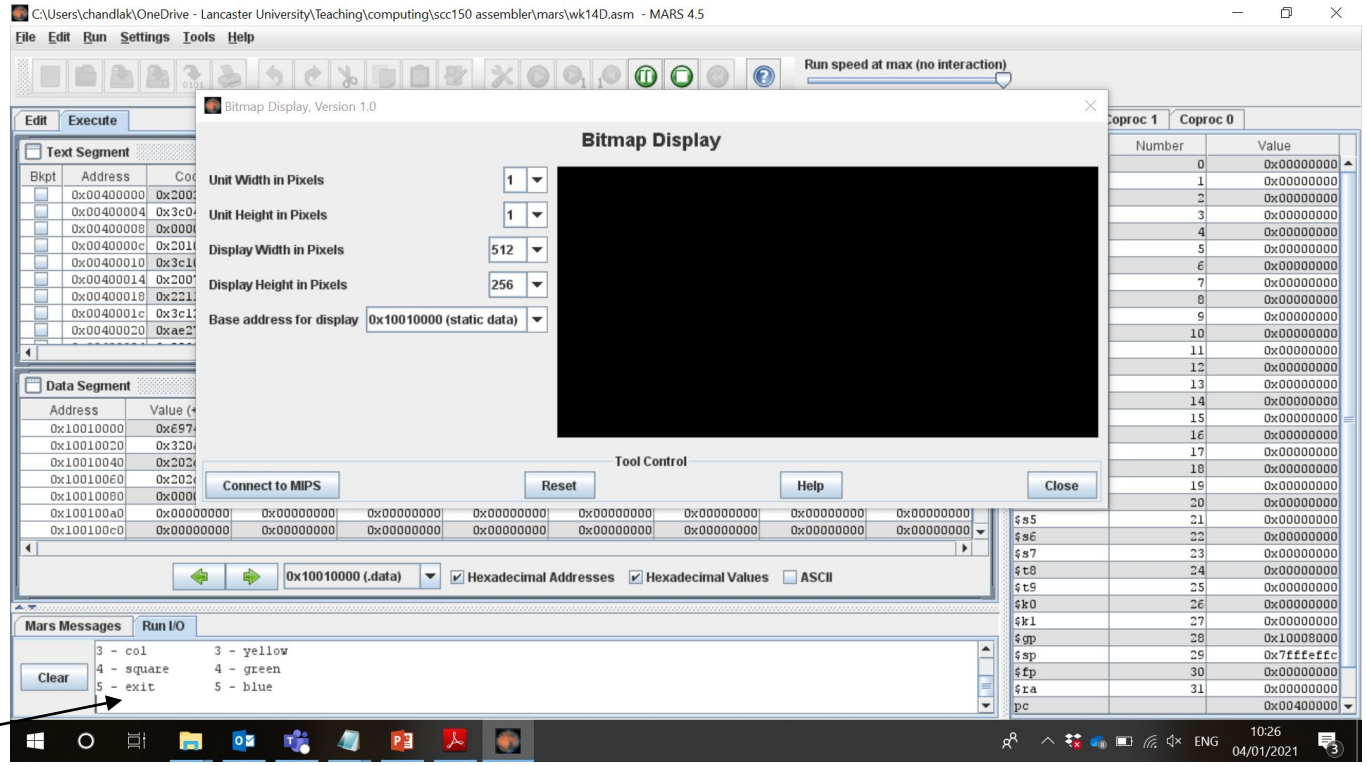
- Using syscall to print to screen
- Using syscall to read an integer
- Interpret instruction
- Drawing on screen
- Implementing procedure
- Music!
- Mark yourself

Syscall - Printing a String

```
.data
#using .data to store the string you want to print
#.asciiz tells it that it has an ascii string and that it ends with a zero
#info is just a label
info:       .asciiz                 "This string has instructions"

.text
#back to main program here (.text)
            addi $v0,$zero,4                    #tell syscall to print string (put code 4 into $v0)
            la $a0, info                        #load the string address (label info) into $a0
                                                #$a0 is the parameter for syscall
            syscall                             #call syscall – it will only print now
```

Syscall - Reading an Integer

- To read an integer, you must put code 5 into syscall ($v0)
- When you call syscall, you can type the integer in using the bottom window
- The result is stored in $v0



here

Interpret Instruction

- You've told the user what to do
- You've read the integer
- Decide what to do with the instruction
- There are two types of instruction:
- Select task
  1. cls (colour background)
  2. Stave (5 horizontal lines)
  3. Note
  4. Exit
- Select colour – there should be several choices of colour

Drawing on screen

Lancaster University

- You must be able to carry out the instruction
- Week 12's practical instructions will help with this if you've forgotten
  - Use heap for display now because print string will use the data section
- CLS
  - Fill in the background of the screen in the correct colour
- Stave
  - Draw 5 equally spaced horizontal lines (in black) using a procedure
- Note
  - Draw a filled square in the appropriate place (in black)
  - Play a note (syscall)
- Exit

CLS – Example (in blue)

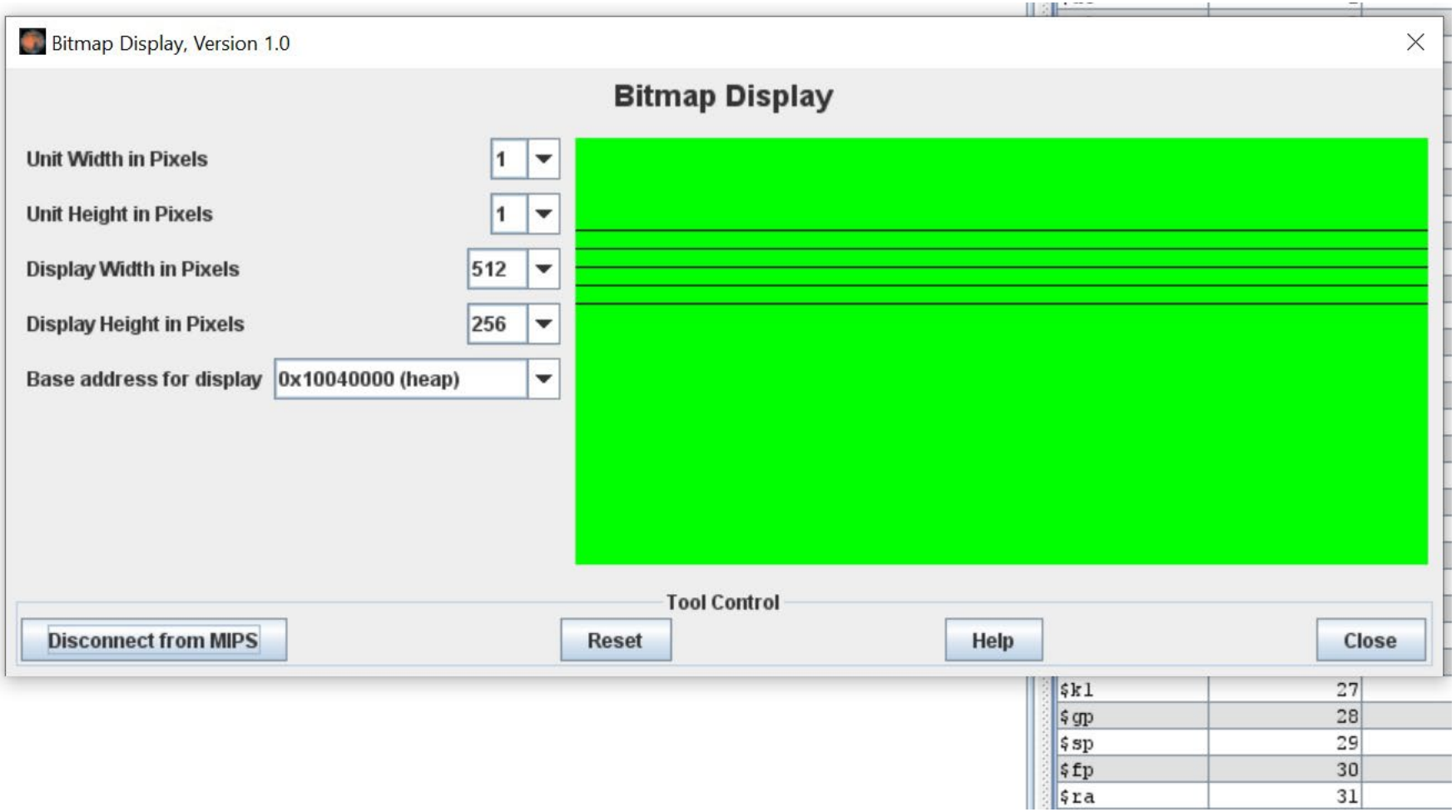
```
lui $s0,0x1004              #bitmap display base address in $s0 (heap)
addi $t8,$zero,0x00ff       #set colour to blue in $t8
addi $t0,$s0,0              #initialise $t0 to base address, will count
lui $s1,0x100C              #end of screen area in $s1

drawPixel:                  #label
        sw $t8,0($t0)       #store colour $t8 in current target address
        addi $t0,$t0,4      #increment $t0 by one word
        bne $t0,$s1,drawPixel  #if haven't reached the target yet, repeat
```

Drawing lines (the stave) - algorithm

- 1 Line:
  - Find start point (based on user input) and paint pixel
  - Add the appropriate number to find the memory location of the next pixel and paint
  - If we have not reached the end of the line, go to 1
  - This should be done in a procedure
- The stave:
  - 5 equally spaced horizontal lines
    - There should be 10 blank lines between each
  - Repeat the procedure for 1 line
  - Work out how to leave a regular space – i.e. equally spaced start points

Example – cls (green), stave (line 55)

```
Options:          Colours:
1 - cls           1 - red
2 - stave         2 - orange
3 - note          3 - yellow
4 - exit          4 - green
                  5 - blue
Select an option
1
Select colour/row/note
4
Options:          Colours:
1 - cls           1 - red
2 - stave         2 - orange
3 - note          3 - yellow
4 - exit          4 - green
                  5 - blue
Select an option
2
Select colour/row/note
55
Options:          Colours:
1 - cls           1 - red
2 - stave         2 - orange
3 - note          3 - yellow
4 - exit          4 - green
                  5 - blue
Select an option
```

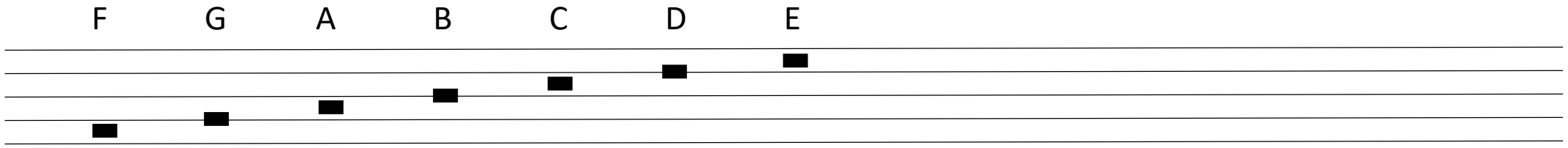


Stave Procedure:
Steps for procedure implementation

- Place parameters in a place where the procedure has access.
- Transfer control to the procedure.
- Acquire the storage resources (e.g. variables, arrays) needed for the procedure.
- Perform the desired task.
- Place the result value in a place where the caller can access it.
- **Return** control to the point of origin.
- j is not the same as jal, you must return!

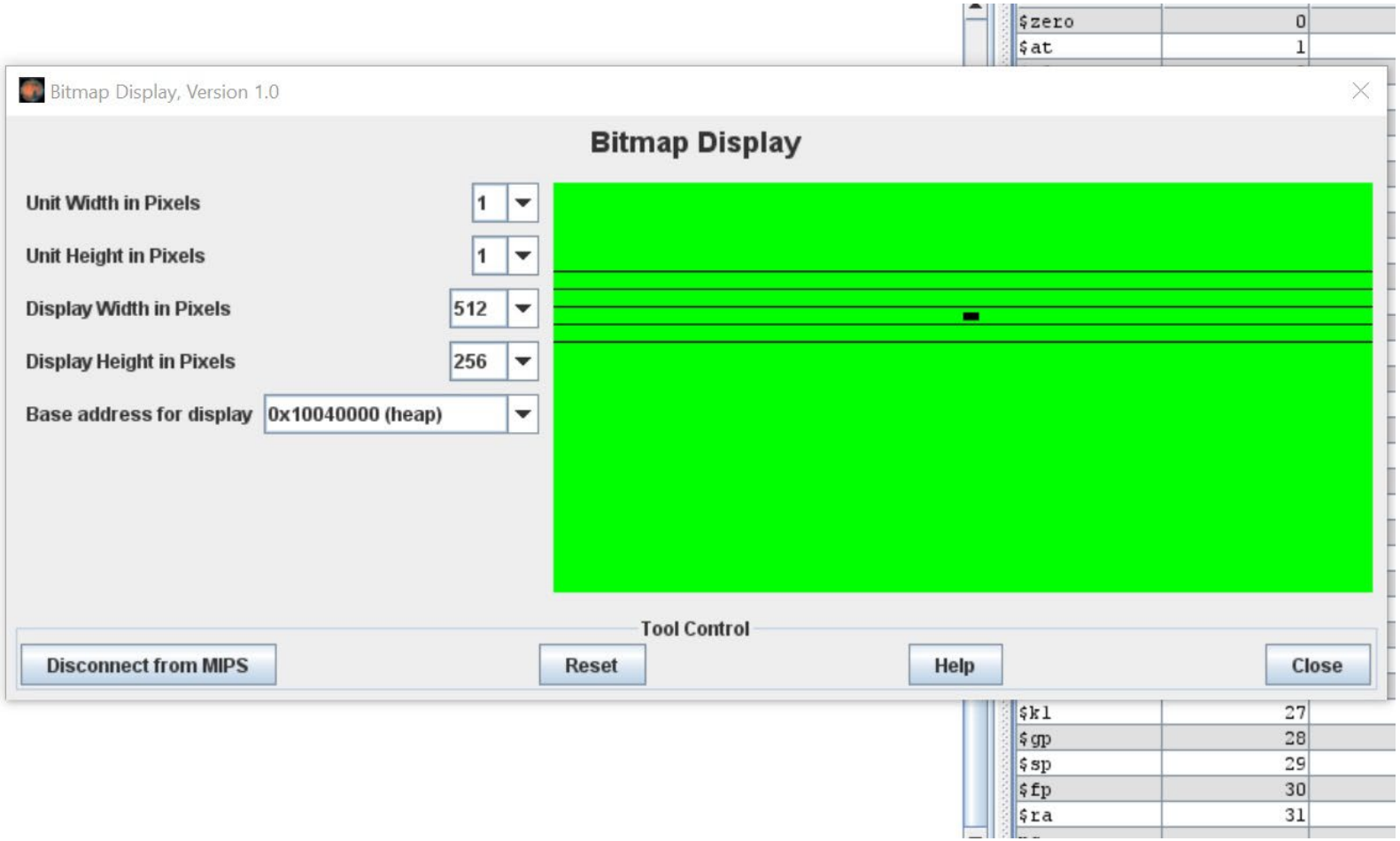
Note – playing a musical note

- Input a CHAR using syscall (a single letter A-G)
  - Don’t press ENTER after a char, as ENTER becomes a char itself
- Draw a small rectangle positioned appropriate to the letter input on the stave (5 horizontal lines). See diagram below.
  - The rectangle should be 8 pixels across and 6 down
- Play the appropriate note (syscall 33)



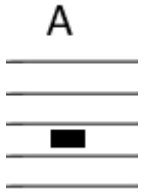
Example – note (user input “A”)

```
Select an option
1
Select colour/row/note
4
Options:        Colours:
1 - cls         1 - red
2 - stave       2 - orange
3 - note        3 - yellow
4 - exit        4 - green
                5 - blue
Select an option
2
Select colour/row/note
55
Options:        Colours:
1 - cls         1 - red
2 - stave       2 - orange
3 - note        3 - yellow
4 - exit        4 - green
                5 - blue
Select an option
3
Select colour/row/note
aOptions:       Colours:
1 - cls         1 - red
2 - stave       2 - orange
3 - note        3 - yellow
4 - exit        4 - green
                5 - blue
Select an option
```

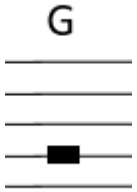


Example – drawing the note - positioning

- You only need to draw one note at a time – so the horizontal positioning of the note is not important (I chose the centre)
- The vertical positioning must match the note input
- In the example, I input A, which must appear in the 3rd space down



- If I chose a different note, it would need to be placed somewhere else, as shown in the “Playing a musical note” slide.
- So, for instance, if the user input G, we’d expect to see the note on the 4th line



- Remember, the position of the stave (lines) can move, so the position of the note must be relative to that

Playing the note

- Use syscall 33. It has the following inputs:
  - $a0 – pitch (the number representing the note, as listed below)
  - $a1 – duration in milliseconds (just pick something sensible and stick with that)
  - $a2 – instrument (anything 0-79 should work)
  - $a3 – volume (be careful it's non-zero!)
- Notes we are using:

F = 65, G = 67, A = 69, B = 71, C = 72, D = 74, E = 76

Getting sound on lab machines

Lancaster University

- Click the speaker icon (top right)
- See the Sound settings panel
- In the Output pane, change the Output Device to "HDMI/DisplayPort – Built-in Audio"
- Change the volume level (System volume pane) to about 1/3 or 2/5

- Of course you need to test, but please be sympathetic to those working around you!

Marking

Lancaster University

- The senior TAs will mark the work offline. They will need:
  - Instructions on anything unusual about your program
  - Your estimated grades – be honest. These are primarily used to detect problems running the code.
- They will be particularly looking out for:
  - Procedures (for the stave)
  - Use of syscalls
- In week 18, the Senior TAs will provide feedback. You should attend this session to get feedback. The TA will:
  - Provide feedback
  - Clarify any queries about the marking