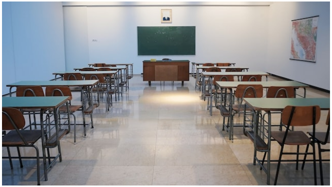
Diverging Assessment for Academic Integrity

David Paul - David.Paul@une.edu.au

The University of New England

2023-07-20

Project Motivation

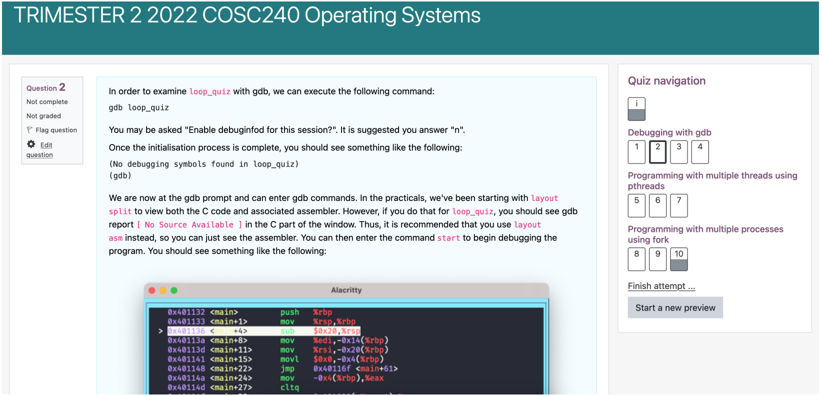


- More assessment moving online
  - Authenticity
  - Online exams
  - Electronic invigilators
- More academic misconduct
  - Plagiarism
  - Collusion
  - Use of AI

Assessment as Learning in Computer Science

- Assignment looks the same to all students
  - Each student has different input data
- Students encouraged to work together
  - Must do assignment on their own
- Reduces plagiarism and collusion!

Assessment as Learning in COSC240 - Practice Quiz



In order to examine `loop_quiz` with gdb, we can execute the following command:

```
gdb loop_quiz
```

You may be asked "Enable debuginfod for this session?". It is suggested you answer "n".

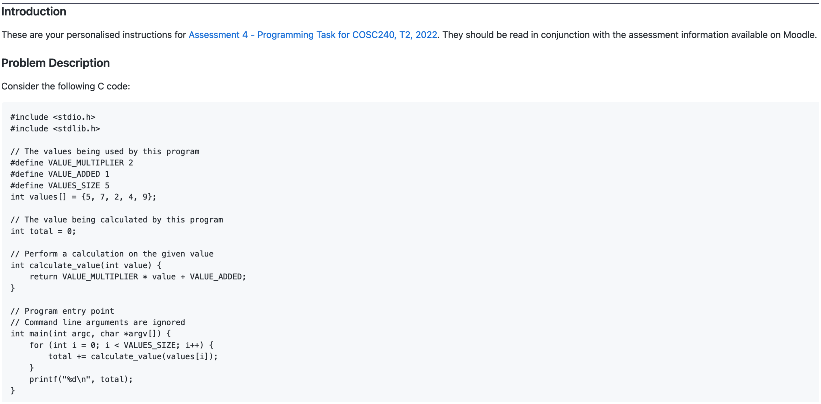
Once the initialisation process is complete, you should see something like the following:

```
(No debugging symbols found in loop_quiz)
(gdb)
```

We are now at the gdb prompt and can enter gdb commands. In the practicals, we've been starting with `layout split` to view both the C code and associated assembler. However, if you do that for `loop_quiz`, you should see gdb report `[ No Source Available ]` in the C part of the window. Thus, it is recommended that you use `layout asm` instead, so you can just see the assembler. You can then enter the command `start` to begin debugging the program. You should see something like the following:

```
   0x401132 <main>         push   %rbp
   0x401133 <main+1>       mov    %rsp,%rbp
 > 0x401136 <main+4>       sub    $0x20,%rsp
   0x40113a <main+8>       mov    %edi,-0x14(%rbp)
   0x40113d <main+11>      mov    %rsi,-0x20(%rbp)
   0x401141 <main+15>      movl   $0x0,-0x4(%rbp)
   0x401148 <main+22>      jmp    0x40116f <main+61>
   0x40114a <main+24>      mov    -0x4(%rbp),%eax
   0x40114d <main+27>      cltq
```

Assessment as Learning in COSC240 - Diverging Assessment



Introduction

These are your personalised instructions for Assessment 4 - Programming Task for COSC240, T2, 2022. They should be read in conjunction with the assessment information available on Moodle.

Problem Description

Consider the following C code:

```
#include <stdio.h>
#include <stdlib.h>

// The values being used by this program
#define VALUE_MULTIPLIER 2
#define VALUE_ADDED 1
#define VALUES_SIZE 5
int values[] = {5, 7, 2, 4, 9};

// The value being calculated by this program
int total = 0;

// Perform a calculation on the given value
int calculate_value(int value) {
    return VALUE_MULTIPLIER * value + VALUE_ADDED;
}

// Program entry point
// Command line arguments are ignored
int main(int argc, char *argv[]) {
    for (int i = 0; i < VALUES_SIZE; i++) {
        total += calculate_value(values[i]);
    }
    printf("%d\n", total);
}
```

Against Collusion and Generative AI

- Collaboration useful
  - Allows students to help each other
- Collusion means doing it more than once
  - Which helps ensure learning
- Formats not understood by generative AI

Thank You

- Diverging assessment through assessment as learning
  - Helps students take responsibility for their own learning
  - Helps maintain academic integrity
  - Works well for COSC240
- Any questions/comments?
  - David.Paul@une.edu.au