

CPSC 240: Computer Organization and Assembly Language

Assignment 01, Fall Semester 2024

CWID: 885024539 Name: Riley Blacklock

1. Download the “CPSC-240 Assignment01.docx” document.
2. Follow the “CPSC-240 Ex01 Hello World.pdf” slide to design a “hello.asm” Assembly program and generate. “hello.o”, “hello.lst”, and “hello” files.
3. Copy and paste the “hello.asm” source code into the document.
4. Follow the “CPSC-240 Ex01 Debugger.pdf” slide to debug the “hello” file.
5. When the program runs to line 15, copy and paste the "Register" window into the document.
6. When the program runs to line 21, copy and paste the "Register" window into the document.
7. When running the "x/14db &text" and "x/s &text" commands, copy and paste the "GDB" window (including the gdb panel) into the document to display the memory results.
8. Save the file in pdf format and submit the pdf file to Canvas before deadline.

[Insert hello.asm source code]

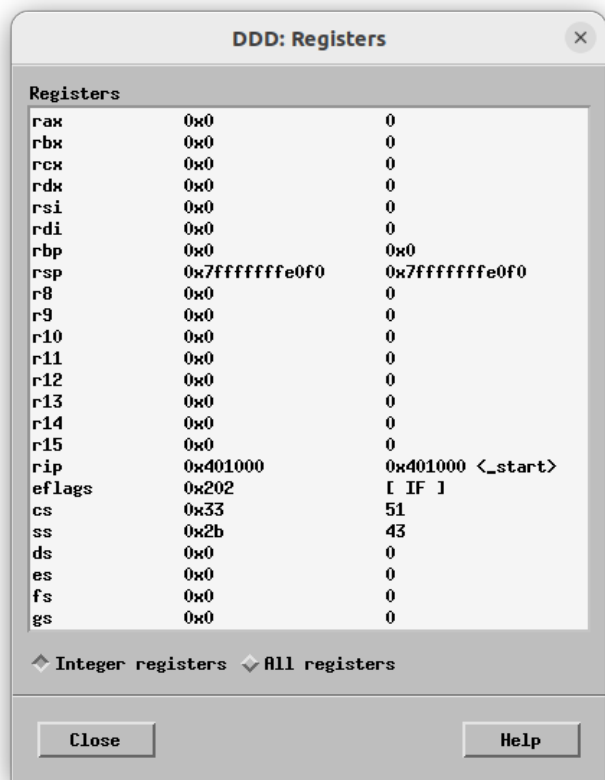
```
; hello.asm
; char text[] = "Hello, World!\n"
; cout << text;

section .data
LF      equ 10
NULL    equ 0
SYS_exit equ 60
EXIT_SUCCESS equ 0
text    db  "Hello, World!", LF, NULL

section .text
global _start
_start:
    mov rax, 1
    mov rdi, 1
    mov rsi, text
    mov rdx, 14
    syscall

    mov rax, SYS_exit
    mov rdi, EXIT_SUCCESS
    syscall
```

[Insert 1st Register window to display initial values]



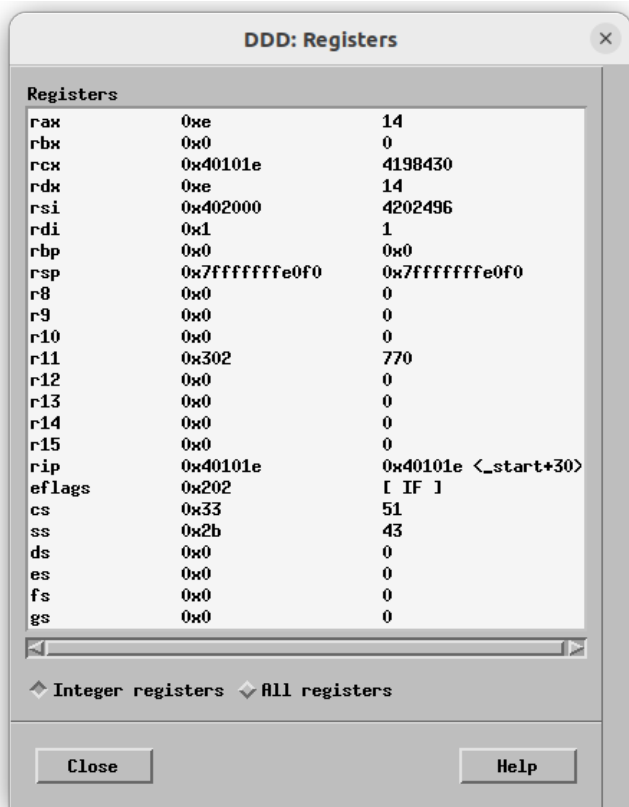
The screenshot shows a window titled "DDD: Registers" with a close button in the top right corner. Inside, there is a table of registers and their values. Below the table, there are two radio buttons: "Integer registers" (which is selected) and "All registers". At the bottom, there are "Close" and "Help" buttons.

Registers		
rax	0x0	0
rbx	0x0	0
rcx	0x0	0
rdx	0x0	0
rsi	0x0	0
rdi	0x0	0
rbp	0x0	0x0
rsp	0x7fffffff0f0	0x7fffffff0f0
r8	0x0	0
r9	0x0	0
r10	0x0	0
r11	0x0	0
r12	0x0	0
r13	0x0	0
r14	0x0	0
r15	0x0	0
rip	0x401000	0x401000 <_start>
eflags	0x202	[IF]
cs	0x33	51
ss	0x2b	43
ds	0x0	0
es	0x0	0
fs	0x0	0
gs	0x0	0

◆ Integer registers ◆ All registers

Close Help

[Insert 2nd Register window here simulated values]



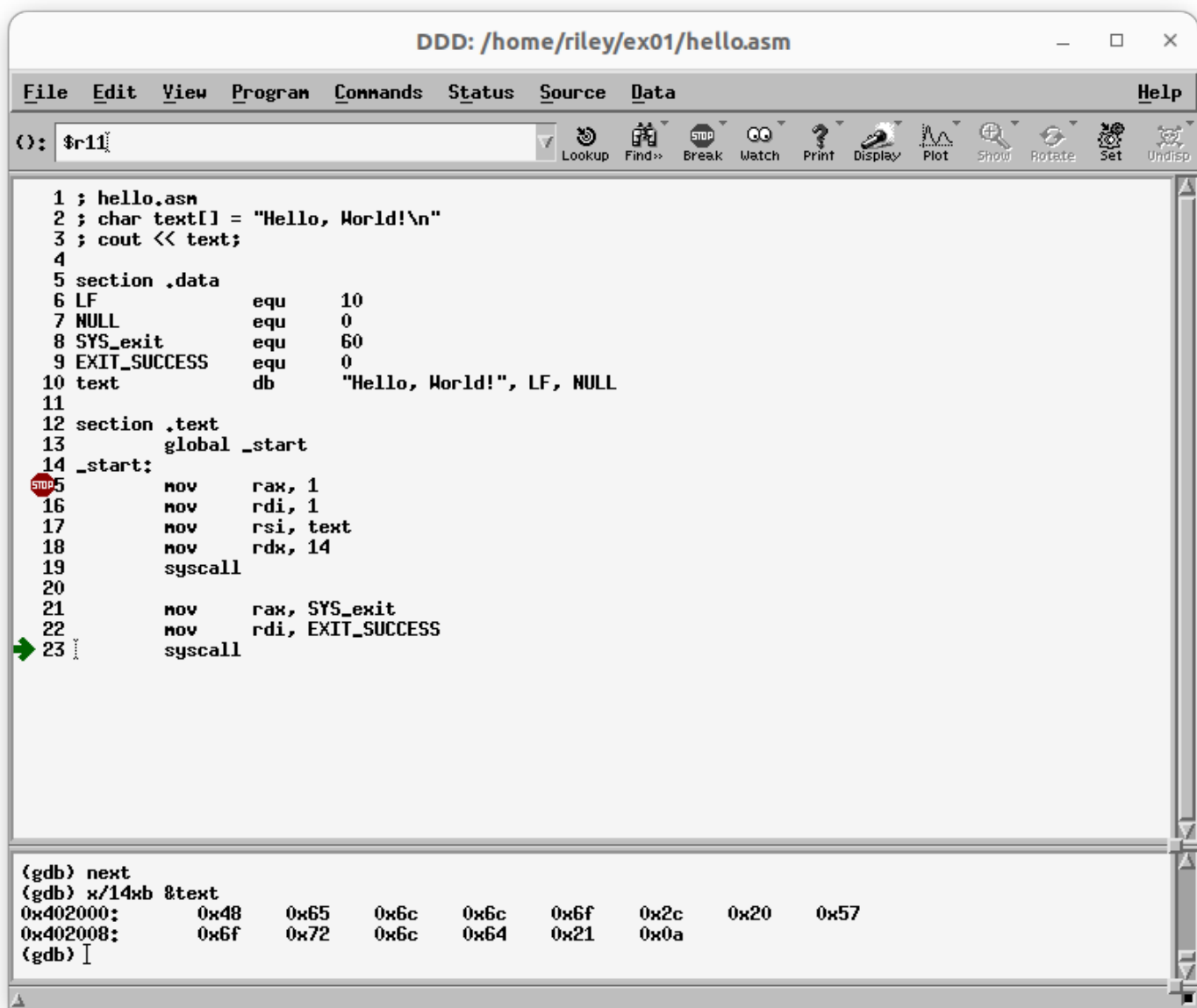
The screenshot shows a window titled "DDD: Registers" with a close button in the top right corner. Inside, there is a table of registers and their values. Below the table, there are two radio buttons: "Integer registers" (which is selected) and "All registers". At the bottom, there are "Close" and "Help" buttons.

Registers		
rax	0xe	14
rbx	0x0	0
rcx	0x40101e	4198430
rdx	0xe	14
rsi	0x402000	4202496
rdi	0x1	1
rbp	0x0	0x0
rsp	0x7fffffff0f0	0x7fffffff0f0
r8	0x0	0
r9	0x0	0
r10	0x0	0
r11	0x302	770
r12	0x0	0
r13	0x0	0
r14	0x0	0
r15	0x0	0
rip	0x40101e	0x40101e <_start+30>
eflags	0x202	[IF]
cs	0x33	51
ss	0x2b	43
ds	0x0	0
es	0x0	0
fs	0x0	0
gs	0x0	0

◆ Integer registers ◆ All registers

Close Help

[Insert GDB window to display the variables]



```
DDD: /home/riley/ex01/hello.asm
File Edit View Program Commands Status Source Data Help
(): $r11
Lookup Find Break Watch Print Display Plot Show Rotate Set Undis

1 ; hello.asm
2 ; char text[] = "Hello, World!\n"
3 ; cout << text;
4
5 section .data
6 LF equ 10
7 NULL equ 0
8 SYS_exit equ 60
9 EXIT_SUCCESS equ 0
10 text db "Hello, World!", LF, NULL
11
12 section .text
13 global _start
14 _start:
15     mov rax, 1
16     mov rdi, 1
17     mov rsi, text
18     mov rdx, 14
19     syscall
20
21     mov rax, SYS_exit
22     mov rdi, EXIT_SUCCESS
23     syscall

(gdb) next
(gdb) x/14xb &text
0x402000: 0x48 0x65 0x6c 0x6c 0x6f 0x2c 0x20 0x57
0x402008: 0x6f 0x72 0x6c 0x64 0x21 0x0a
(gdb) [
```