COMP1562 Logbook (Week 2)

Basic Information

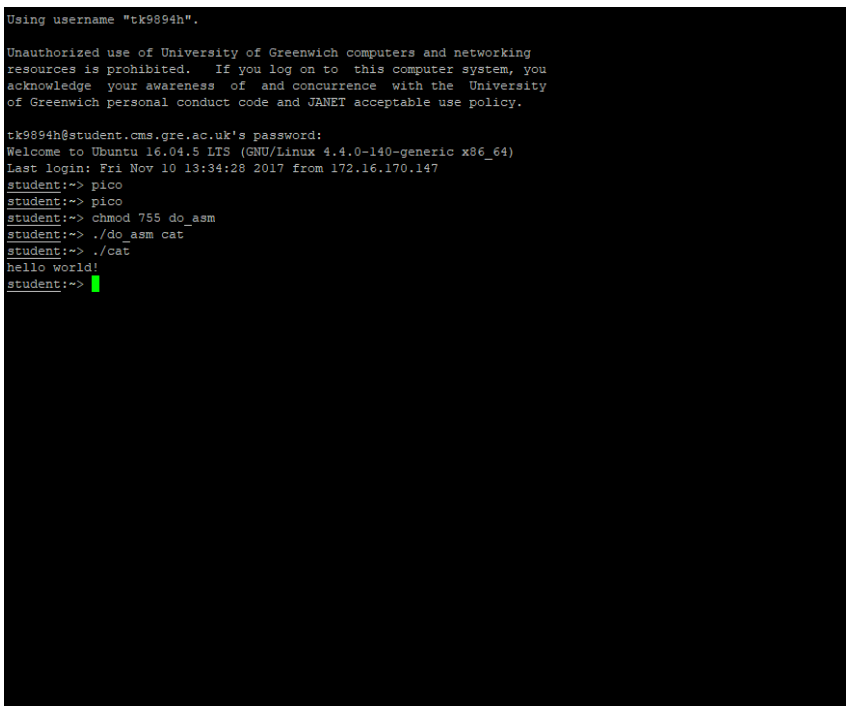
1.1 Student name	**Trevor Kiggundu (001001720)**
1.2 Who did you work with? Name and/or id	**Maruf Hoque (001006731)**
1.3 Which lab topic does this document relate to?	System Bit Processors and System Shells
1.4 How well do you feel you have done?	I have completed the exercise and am totally satisfied with my work.
1.5 Briefly explain your answer to question 1.4	My group and I were able to successfully follow and complete the tasks. Proof of that is shown below.

Annotated screen shots demonstrating what you have achieved:

TASK 1:

Exercise 1:

Screenshot showing successful completion of exercise 1, which was to run the code to display "Hello World". The saved program is named 'cat':



Exercise 2:

This exercise familiarized us with the use of the debugger.
Shown below is the creation of the second program named ‘cat2’:

```
Using username "tk9894h".

Unauthorized use of University of Greenwich computers and networking
resources is prohibited.   If you log on to  this computer system, you
acknowledge  your awareness  of  and concurrence  with the  University
of Greenwich personal conduct code and JANET acceptable use policy.

tk9894h@student.cms.gre.ac.uk's password:
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-140-generic x86_64)
Last login: Fri Nov 10 13:34:28 2017 from 172.16.170.147
student:~> pico
student:~> pico
student:~> chmod 755 do_asm
student:~> ./do_asm cat
student:~> ./cat
hello world!
student:~> pico
student:~> ./do_asm cat2
student:~> ./cat2
student:~> gdb cat2
GNU gdb (Ubuntu 7.11.1-0ubuntu1~16.5) 7.11.1
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from cat2...(no debugging symbols found)...done.
(gdb)
(gdb)
(gdb) q
student:~> info registers
student:~>
```

After debugging, shown below is the use of the ‘nexti’ command to step through the program and ensure that the correct values are in each register. The final answer is 6, shown in register $3:

```
Using username "tk9894h".

Unauthorized use of University of Greenwich computers and networking
resources is prohibited.   If you log on to  this computer system, you
acknowledge  your awareness  of  and concurrence  with the  University
of Greenwich personal conduct code and JANET acceptable use policy.

tk9894h@student.cms.gre.ac.uk's password:
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-141-generic x86_64)
Last login: Wed Jan 23 17:25:52 2019 from 172.16.170.191
student:~> gdb cat2
GNU gdb (Ubuntu 7.11.1-0ubuntu1~16.5) 7.11.1
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from cat2...(no debugging symbols found)...done.
(gdb) break main
Breakpoint 1 at 0x80483e0
(gdb) run
Starting program: /home/tk9894h/cat2

Breakpoint 1, 0x080483e0 in main ()
(gdb) print $ax
$1 = 7612
(gdb) nexti
0x080483e6 in main ()
(gdb) print $ax
$2 = 2
(gdb) nexti
0x080483ed in main ()
(gdb) print $ax
$3 = 6
(gdb) print $NUMBER2
$4 = void
(gdb)
```

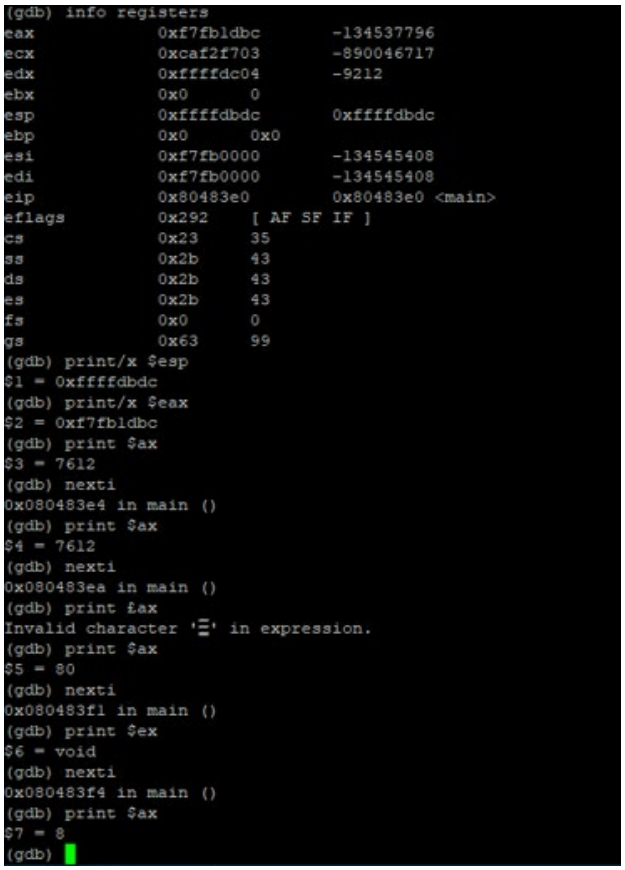
Exercise 3:

Shown below is the modified program to change to the division of two numbers:

Code:

```
section .data
NUMBER1:        dw 80
NUMBER2:        dw 10

section .text
        global main

main:

        mov dx,0
        mov ax,[NUMBER1]
        mov cx,[NUMBER2]
        div cx

        mov eax,1
        mov ebx,0
        int 80h
```

After debugging, shown below is the use of the 'nexti' command to step through the program and ensure that the correct values are in each register. The final answer is 8, shown in register $7:



TASK 2:

This task asked us to produce a program that can calculate the area of a trapezoid.

$$\boldsymbol{result = \frac{a+b}{2} \times h}$$

Expected Result:
= 3+7/2 *4
=5*4
=**20**
Shown below is the creation of the program, as well as running the debugger:

```
Unauthorized use of University of Greenwich computers and networking
resources is prohibited.   If you log on to  this computer system, you
acknowledge  your awareness  of  and concurrence  with the  University
of Greenwich personal conduct code and JANET acceptable use policy.

mh9487h@student.cms.gre.ac.uk's password:
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-141-generic x86_64)
Last login: Sun Jan 27 12:47:23 2019 from 172.16.170.16
student:~> pico
student:~> chmod 755 do_asm
student:~> ./do_asm example3
student:~> ./do_asm example3
student:~> ./example3
student:~> gbd example3
gbd: Command not found.
student:~> gdb example3
GNU gdb (Ubuntu 7.11.1-0ubuntu1~16.5) 7.11.1
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from example3...(no debugging symbols found)...done.
(gdb) break main
```

After debugging, shown below is the use of the 'nexti' command to step through the program and ensure that the correct values are in each register. The final answer is 20, shown in register $7:

```
Breakpoint 1, 0x080483e0 in main ()
(gdb) nexti
0x080483e6 in main ()
(gdb) print $ax
$1 = 3
(gdb) nexti
0x080483ed in main ()
(gdb) print $ax
$2 = 10
(gdb) nexti
0x080483f1 in main ()
(gdb) print $bx
$3 = 2
(gdb) nexti
0x080483f5 in main ()
(gdb) print $ax
$4 = 10
(gdb) nexti
0x080483f8 in main ()
(gdb) print $ax
$5 = 5
(gdb) nexti
0x080483ff in main ()
(gdb) print $bx
$6 = 4
(gdb) nexti
0x08048402 in main ()
(gdb) print $ax
$7 = 20
```

Shown below is the code we used to calculate the answer:

Group ID (nr):

18

Task 2 - Real program assembly

You will need to write the mentioned pieces of code to implement the following arithmetic operation:

result=((a+b)/2)*h

Please mind that only instructions comprising variables declaration and all arithmetic instructions required to calculate the equation are to be provided. Rest of the code (like program termination instruction etc.) will be added automatically.

Your variables declarations:

```
NUMBER1: dw 3
NUMBER2: dw 7
NUMBER3: dw 2
NUMBER4: dw 4
result:  dw 0
```

Your code to calculate the equation:

```
mov ax, [NUMBER1]
add ax, [NUMBER2]
mov dx, 0
mov bx, [NUMBER3]
div bx
mov bx, [NUMBER4]
mul bx
mov [result], ax
```

Personal Reflection:

This weeks' lab was slightly more challenging than last weeks', but it was still very enjoyable to complete. The lab required us to use the 'Putty' application, one that we had not used since first year, so it took us a couple of days to find our old passwords. We were already familiar with pseudo program execution from the previous weeks' task, so we were able to jump in right away. The toughest part of this lab was not reading the instructions carefully. Fetching instructions from the next register is done by using the 'nexti' command. However, I kept accidentally using the 'next' command, which moves on to the next breakpoint. This kept causing the program to terminate as we only had one breakpoint. We also had problems executing the Task 2 assembly program at first, as it was running in scriptcheck and made sense logically on paper. In the end, we were able to follow the previous examples again and apply it to finally reach the correct solution. I am glad that I experienced this lab session and 100% believe that is has made me a better programmer and given me a better understanding of system shells.