

execv עבודה בלינוקס ריבוי תהליכים

חלק א : תרגול עבודה עם דיבאגר

במקרה שתעודת הזהות שלכם מסתיימת בספרת ביקורת 1-3 עבדו על באג "א"
במקרה שתעודת הזהות שלכם מסתיימת בספרת ביקורת 4-6 עבדו על באג "ב"
במקרה שתעודת הזהות שלכם מסתיימת בספרת ביקורת 7-0 עבדו על באג "ג"

במקרה ויש מספר חברים בקבוצה הגישו את הבאגים הרלוונטים לכל החברים.

באג א : צור תוכנית הנופלת עקב חלוקה באפס.

באג ב : צור תוכנית הנופלת עקב גישה לא חוקית לזכרון. (זכרון לא מאותחל)

באג ג : צור תוכנית הנופלת עקב רקורסיה אין סופית.

לגבי הבאג המתאים לכם.

1. הדגימו קומפילציה של התוכנית עם וברי debug
2. הריצו את התוכנית (תנו לה ליפול)
3. הדגימו את הצעדים הדרושים לקבלת core
4. פתחו את coren בעזרת gdb - הדגימו את ההבדלים בפלט (בעזרת copy paste או צילום מסך) בין קלט עם וברי דיבאג.
5. הדגימו את מציאת השורה הרלוונטית של הנפילה בקוד בעזרת core.
6. הדגימו פתיחה של coren בעזרת debugger גרפי - לדוגמא ddd.

```
C Bug_c.c U X
OperatingSystems_Ex1 > C Bug_c.c > main()
1  #include <stdio.h>
2
3  int plusOne(){
4      return plusOne()+1;
5  }
6
7  int main(){
8      plusOne();
9      return 0;
10 }
```

We will start the project by creating the 3rd bug (a program that fails due to infinite recursion). As we can see in the snapshot, we created a program that adds 1 each time without ever ending.

```
maya@maya:~/Desktop/OperatingSystems_Ex1$ ls
Bug_c.c
maya@maya:~/Desktop/OperatingSystems_Ex1$ gcc Bug_c.c -o Bug_c
maya@maya:~/Desktop/OperatingSystems_Ex1$ ls
Bug_c  Bug_c.c
maya@maya:~/Desktop/OperatingSystems_Ex1$ ./Bug_c
Segmentation fault (core dumped)
maya@maya:~/Desktop/OperatingSystems_Ex1$
```

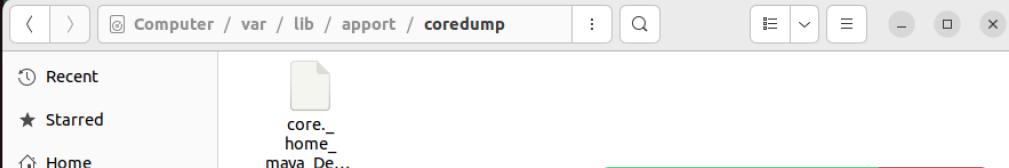
The second step is to compile our program without debug (note how Bug_c was added to the folder after running the compilation line).

After having done this we will try to compile our program with debug as follows:

```
maya@maya:~/Desktop/OperatingSystems_Ex1$ gcc -o Bug_c_debug -g Bug_c.c
maya@maya:~/Desktop/OperatingSystems_Ex1$ ./Bug_c_debug
Segmentation fault (core dumped)
maya@maya:~/Desktop/OperatingSystems_Ex1$ ls
Bug_c  Bug_c.c  Bug_c_debug
maya@maya:~/Desktop/OperatingSystems_Ex1$
```

In order to get core, we have to run “ulimit -c unlimited” and then the program and we can find the core dump under /var/lib/apport/coredump. We can see that there is only 1 core, we will copy it to our folder and continue from there.

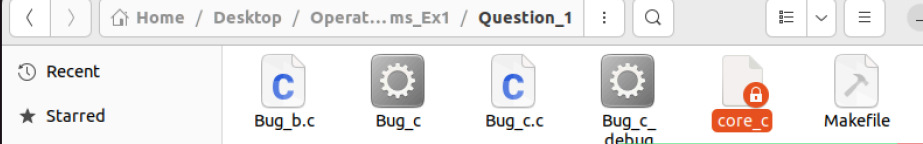
```
maya@maya:~/Desktop/OperatingSystems_Ex1/Question_1$ make all
gcc -Wall Bug_c.c -o Bug_c
gcc -Wall -g Bug_c.c -o Bug_c_debug
maya@maya:~/Desktop/OperatingSystems_Ex1/Question_1$ ls
Bug_b.c  Bug_c  Bug_c.c  Bug_c_debug  Makefile
maya@maya:~/Desktop/OperatingSystems_Ex1/Question_1$ ./Bug_c
Segmentation fault (core dumped)
maya@maya:~/Desktop/OperatingSystems_Ex1/Question_1$ ./Bug_c_debug
Segmentation fault (core dumped)
maya@maya:~/Desktop/OperatingSystems_Ex1/Question_1$ ulimit -c unlimited
maya@maya:~/Desktop/OperatingSystems_Ex1/Question_1$ ./Bug_c
Segmentation fault (core dumped)
```



In here we can see that we copied the core into our current folder, changed its name into “core_c” in order to make our folder as well as our future code neater, and run “gdb”.

```
maya@maya:~/Desktop/OperatingSystems_Ex1/Question_1$ ./Bug_c
Segmentation fault (core dumped)
maya@maya:~/Desktop/OperatingSystems_Ex1/Question_1$ gdb
GNU gdb (Ubuntu 12.0.90-0ubuntu1) 12.0.90
Copyright (C) 2022 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:
<https://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".
(gdb)
```



First we tried to open our core_c without debug and got the results shown on the first screenshot. It shows that the program ended with a segmentation fault and points us to the plusOne() function (suggesting there is something wrong about it). Additionally it mentions that no debugging symbols were mentioned in Bug_c.

```
maya@maya:~/Desktop/OperatingSystems_Ex1/Question_1$ make core_bug_c
gdb -c core_c Bug_c
GNU gdb (Ubuntu 12.0.90-0ubuntu1) 12.0.90
Copyright (C) 2022 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:
<https://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 Bug_c...
(No debugging symbols found in Bug_c)
[New LWP 5164]
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Core was generated by './Bug_c'.
Program terminated with signal SIGSEGV, Segmentation fault.
#0  0x000055b125c6e136 in plusOne ()
(gdb) exit
```

Here, in the second screenshot, we can see what happens when running gdb with debugging. Note how here it says “reading symbols from ...” where in the previous run (without the debugger) it said “No debugging symbols”.

```
maya@maya:~/Desktop/OperatingSystems_Ex1/Question_1$ make Bug_c_debug
gcc -Wall -g3 Bug_c.c -o Bug_c_debug
maya@maya:~/Desktop/OperatingSystems_Ex1/Question_1$ ls
Bug_b.c Bug_c.c Bug_c_debug core_c Makefile
maya@maya:~/Desktop/OperatingSystems_Ex1/Question_1$ make core_bug_c_debug
gdb ./Bug_c_debug
GNU gdb (Ubuntu 12.0.90-0ubuntu1) 12.0.90
Copyright (C) 2022 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:
<https://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 ./Bug_c_debug...
(gdb) █
```

After opening that, we need to run the debugger, we will do that by pressing “r”. When doing so, it tells us that it is running our program and finds the exact line where the problem occurred, in our case it points at line 4

```
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./Bug_c_debug...
(gdb) r
Starting program: /home/maya/Desktop/OperatingSystems_Ex1/Question_1/Bug_c_debug
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
```

```
Program received signal SIGSEGV, Segmentation fault.
```

```
0x000055555555136 in plusOne () at Bug_c.c:4
```

```
4       return plusOne()+1;
```

```
(gdb) list
```

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

```
2
```

```
3       int plusOne(){
```

```
4           return plusOne()+1;
```

```
5       }
```

```
6
```

```
7       int main(){
```

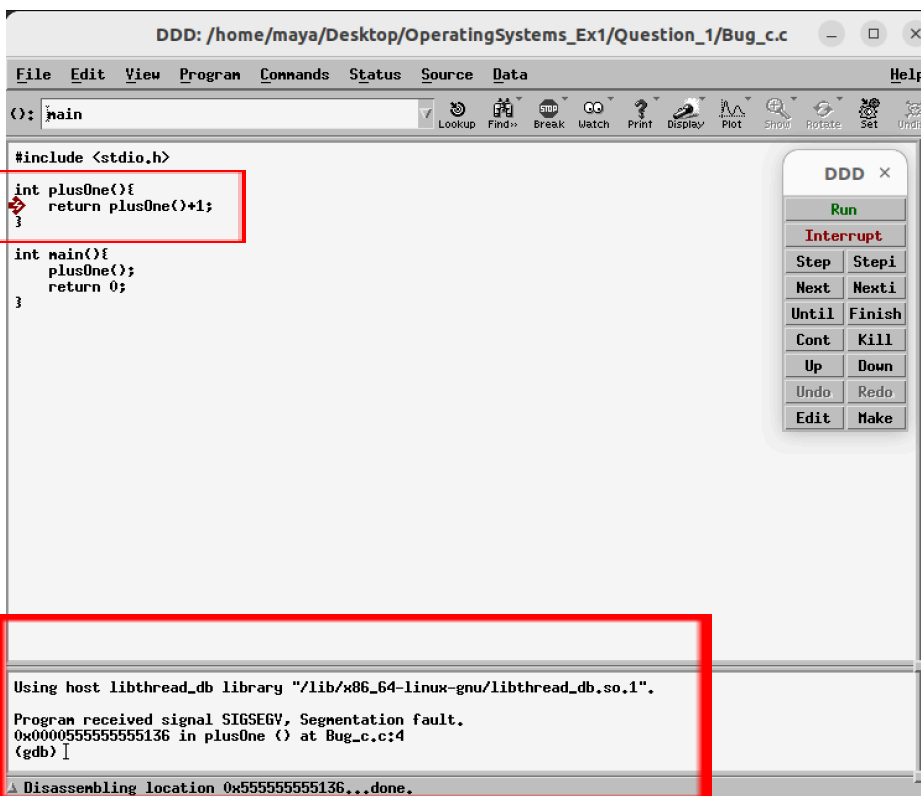
```
8           plusOne();
```

```
9           return 0;
```

```
10      }
```

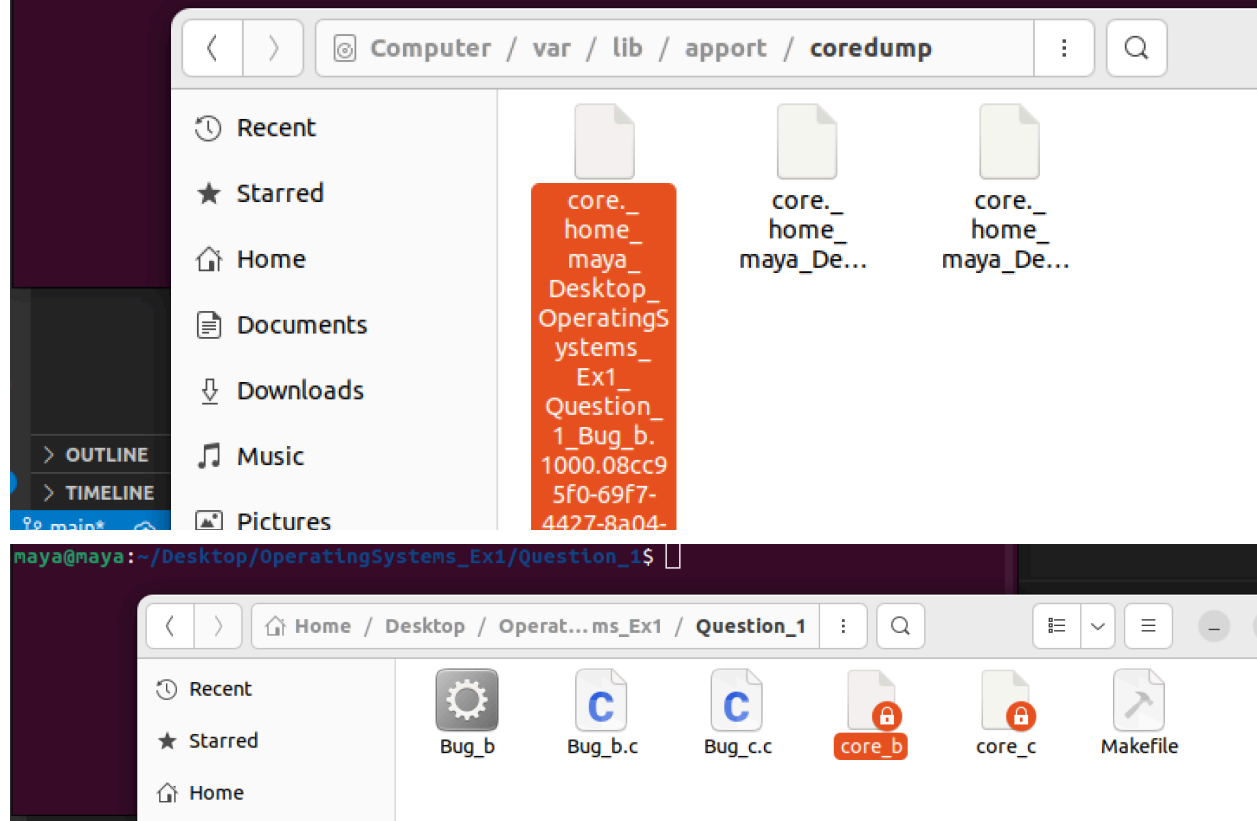
```
(gdb)
```

You are screen sharing



Bug B

```
maya@maya:~/Desktop/OperatingSystems_Ex1/Question_1$ ulimit -c unlimited
maya@maya:~/Desktop/OperatingSystems_Ex1/Question_1$ make Bug_b
make: 'Bug_b' is up to date.
maya@maya:~/Desktop/OperatingSystems_Ex1/Question_1$ ./Bug_b
Segmentation fault (core dumped)
maya@maya:~/Desktop/OperatingSystems_Ex1/Question_1$
```



```
maya@maya:~/Desktop/OperatingSystems_Ex1/Question_1$ make core_bug_b
gdb -c core_b Bug_b
GNU gdb (Ubuntu 12.0.90-0ubuntu1) 12.0.90
Copyright (C) 2022 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:
<https://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 Bug_b...
(No debugging symbols found in Bug_b)
[New LWP 6631]
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Core was generated by './Bug_b'.
Program terminated with signal SIGSEGV, Segmentation fault.

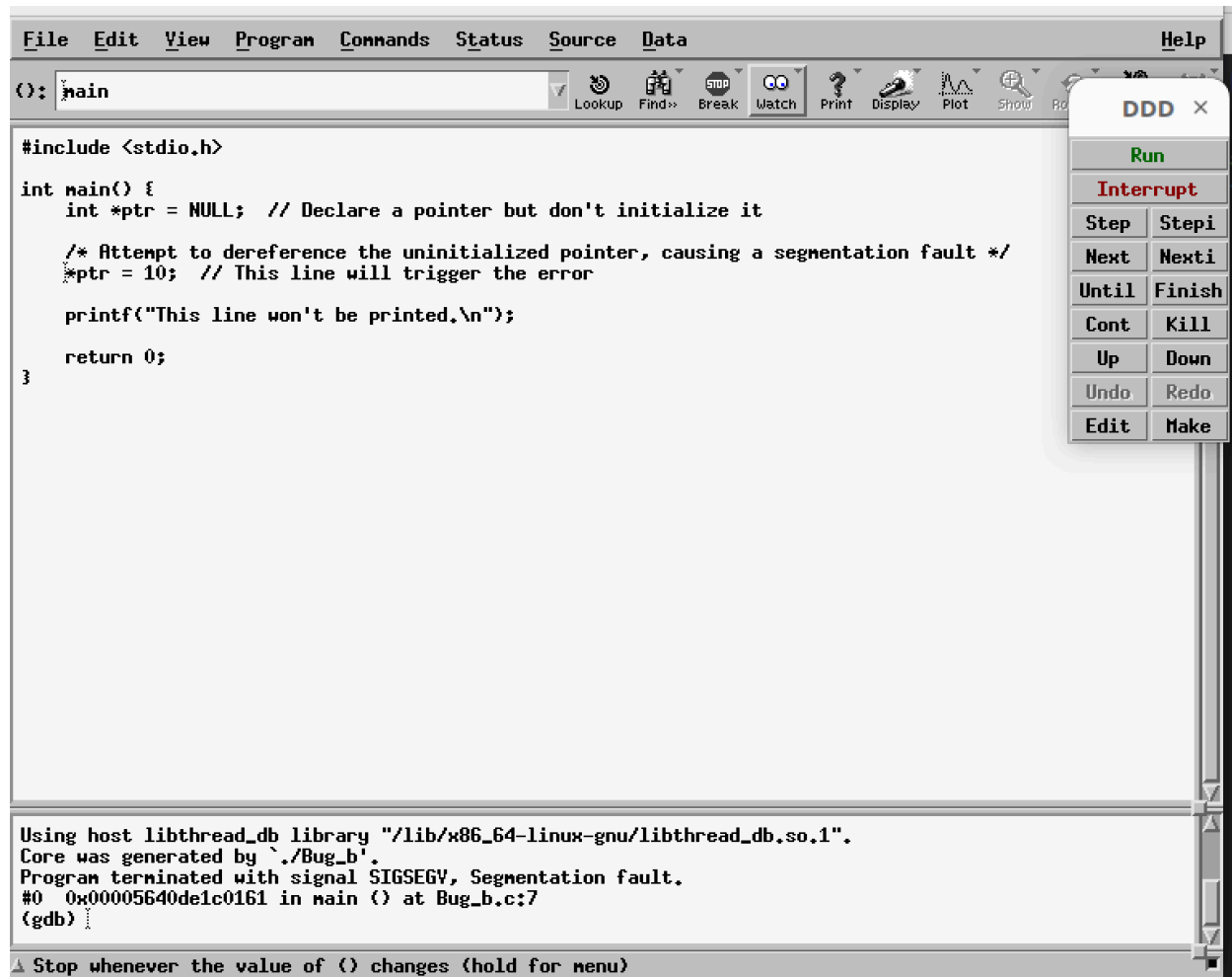
< > Home / Desktop / Operat...ms_Ex1 / Question_1 : Q [ ] [ ] [ ]
Recent
★ Starred
Bug_b Bug_b.c Bug_c.c core_b core_c Makefile
```

```
maya@maya:~/Desktop/OperatingSystems_Ex1/Question_1$ make Bug_b_debug
gcc -Wall -g3 Bug_b.c -o Bug_b_debug
maya@maya:~/Desktop/OperatingSystems_Ex1/Question_1$ ls
Bug_b Bug_b.c Bug_b_debug Bug_c.c core_b core_c Makefile
maya@maya:~/Desktop/OperatingSystems_Ex1/Question_1$ make core_bug_b_debug
gdb ./Bug_b_debug
GNU gdb (Ubuntu 12.0.90-0ubuntu1) 12.0.90
Copyright (C) 2022 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:
<https://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 ./Bug_b_debug...
(gdb) r
Starting program: /home/maya/Desktop/OperatingSystems_Ex1/Question_1/Bug_b_debug
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Program received signal SIGSEGV, Segmentation fault.
0x0000555555555161 in main () at Bug_b.c:7
7      *ptr = 10; // This line will trigger the error
```

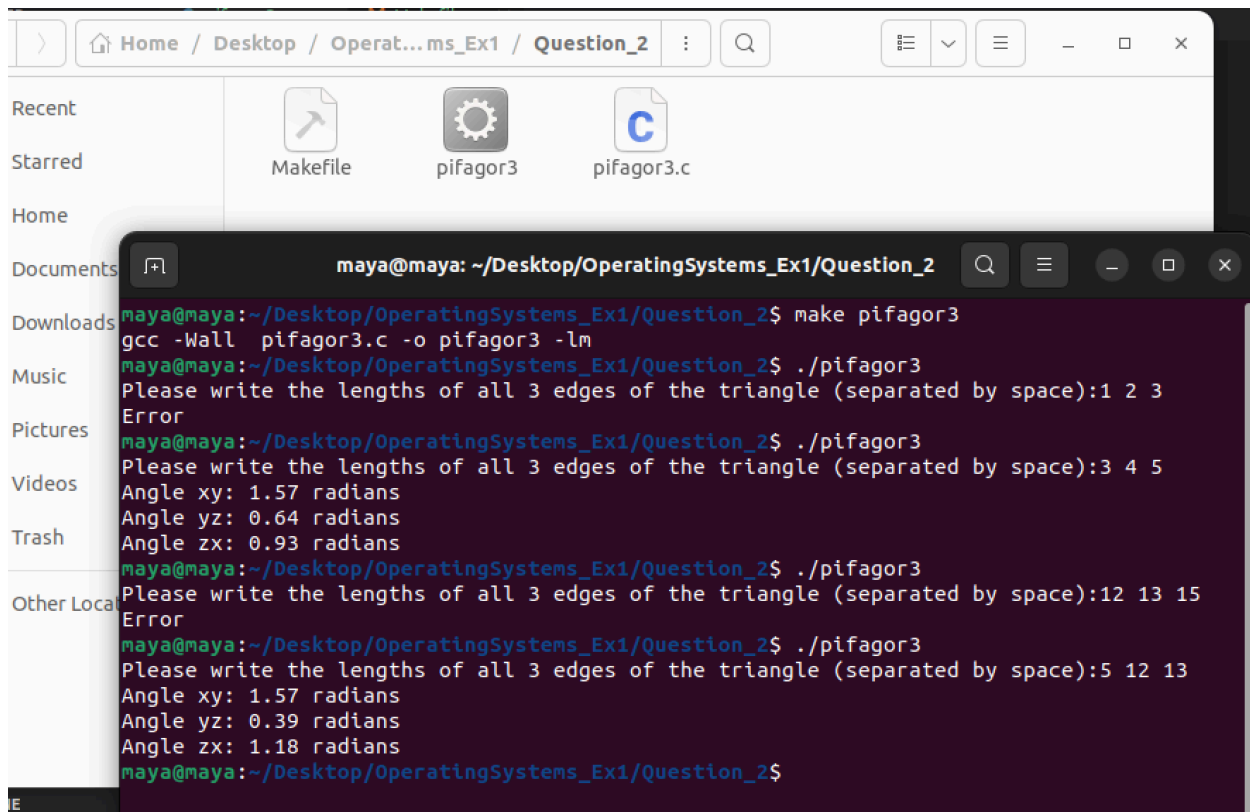
```
Program received signal SIGSEGV, Segmentation fault.
0x0000555555555161 in main () at Bug_b.c:7
7      *ptr = 10; // This line will trigger the error
(gdb) list
2
3      int main() {
4          int *ptr = NULL; // Declare a pointer but don't initialize it
5
6          /* Attempt to dereference the uninitialized pointer, causing a segmentation
fault */
7          *ptr = 10; // This line will trigger the error
8
9          printf("This line won't be printed.\n");
10
11         return 0;
(gdb) |
```



```
mayamaya:~/Desktop/OperatingSystems_Ex1/Question_1$ ddd ./Bug_b_debug core_b
Warning: Cannot convert string "-*-helvetica-medium-r-*-*-*120-*-*-*-*iso8859-*" to type FontStruct
(Annoyed? Try 'Edit->Preferences->General->Suppress X Warnings'!)
Warning: Cannot convert string "-*-helvetica-medium-r-*-*-*100-*-*-*-*iso8859-*" to type FontStruct
Warning: Cannot convert string "-*-lucidatypewriter-medium-r-*-*-*120-*-*-*-*iso8859-*" to type FontStruct
Warning: Cannot convert string "-*-lucidatypewriter-bold-r-*-*-*120-*-*-*-*iso8859-*" to type FontStruct
Warning: Cannot convert string "-*-helvetica-bold-r-*-*-*120-*-*-*-*iso8859-*" to type FontStruct
Warning: Cannot convert string "-*-helvetica-medium-*-*-*120-*-*-*-*iso8859-*" to type FontStruct
Warning: Cannot convert string "-*-helvetica-bold-*-*-*120-*-*-*-*iso8859-*" to type FontStruct
Warning: Cannot convert string "-*-helvetica-bold-r-*-*-*180-*-*-*-*iso8859-*" to type FontStruct
Warning: Cannot convert string "-*-symbol-*-*-*120-*-*-*-*adobe-*" to type FontStruct
```

חלק ב : עבודה עם ספרייה

צרו תוכנית הקולטת מהשתמש 3 אורכי צלעות ובודקת אם מדובר בשלשה פיטגורית.
אם מדובר בשלשה פיתגורית התוכנית תדפיס את הזוויות במשולש - בראדיאנים
אם לא מדובר בשלשה פיתגורית התוכנית תדפיס Error בשורה נפרדת ותצא.
יש להשתמש בספרייה המתמטית הסטדרטית (!) כלומר -lm. על מנת להשתמש בפונקציות (asin) ודומות לה.



```
maya@maya: ~/Desktop/OperatingSystems_Ex1/Question_2
maya@maya:~/Desktop/OperatingSystems_Ex1/Question_2$ make pifagor3
gcc -Wall pifagor3.c -o pifagor3 -lm
maya@maya:~/Desktop/OperatingSystems_Ex1/Question_2$ ./pifagor3
Please write the lengths of all 3 edges of the triangle (separated by space):1 2 3
Error
maya@maya:~/Desktop/OperatingSystems_Ex1/Question_2$ ./pifagor3
Please write the lengths of all 3 edges of the triangle (separated by space):3 4 5
Angle xy: 1.57 radians
Angle yz: 0.64 radians
Angle zx: 0.93 radians
maya@maya:~/Desktop/OperatingSystems_Ex1/Question_2$ ./pifagor3
Please write the lengths of all 3 edges of the triangle (separated by space):12 13 15
Error
maya@maya:~/Desktop/OperatingSystems_Ex1/Question_2$ ./pifagor3
Please write the lengths of all 3 edges of the triangle (separated by space):5 12 13
Angle xy: 1.57 radians
Angle yz: 0.39 radians
Angle zx: 1.18 radians
maya@maya:~/Desktop/OperatingSystems_Ex1/Question_2$
```

An important note is that if we tried compiling our program without the -lm at the end it would have a problem as it won't recognise the math.h functions used (such as M_PI, pow(x,y) and asin()) in this case).

חלק ג : כתיבת ספריה

כתוב תוכנית המיצרת מקודד פוליאלפבטי.
למקודד יש 4 פונקציות

```
;(void * createCodec (char key[62]
```

התוכנית מקבלת את הצופן לאורך 62 תווים המתאר לאיזה אות כל אות תמופה. תחילה 26 אותיות קטנות, לאחר מכן 26 אותיות גולות ולאחר מכן 10 ספרות (מ 1 עד 0) לדוגמא המפתח הבא יצור צופן קיסר של +3
defghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890abc
במידה והמפתח שמתקבל לא חוקי (לדוגמא 2 אותיות ממופות לאותו תו - התוכנית תחזיר null אחרת התוכנית תאלקץ ותחזיר את המבנה של המקודד שיועבר לפונקציות הבאות (אתם רשאים להגדיר אותו כפי שתרצו)

```
;(int encode(char * textin, char * textout, int len, void * codec  
;(int decode(char * textin, char * textout, int len, void * codec  
2 פונקציות המציפות או מפענחות את textin לתוך textout בעזרת ה codec (שאותחל בעזרת createCodec  
Len - הוא אורך המחרוזת  
textin, textout זה פשוט באפרים שאולקצו מראש
```

בכל מקרה של הצלחה הפונקציה תחזיר את מספר הבטים שפוענחו או קודדו.
בכל מקרה של כשלון המספר המוחזר יהיה שונה מlen או יהיה -1 (לדוגמא אם textin הוא null)

```
;(Void freecodec(void * codec  
התוכנית תקמפל את המקודד לספריה  
השתמש בספריה על מנת לכתוב 2 תוכניות אחרות.  
encode src_file dst_file  
decode src_file dst_file  
שתי התוכניות מקבלות שם של קבצים כפרמטרים ומקודדות או מפענחות את הקובץ קלט בעזרת הספריה.  
התוכניות רושמות את הפלט לקובץ הפלט.
```

```
OperatingSystems_Ex1 > Question_3 > Output.txt
1
2

Input.txt
1 1234567890 hello my name is Israel

maya@maya: ~/Desktop/OperatingSystems_Ex1/Question_3
maya@maya:~/Desktop/OperatingSystems_Ex1/Question_3$ make clean
rm -f libCodec.o mainEncode.o mainDecode.o encode decode libCodec.so
maya@maya:~/Desktop/OperatingSystems_Ex1/Question_3$ ls
Input.txt  libCodec.h  mainEncode.c  Output.txt
libCodec.c  mainDecode.c  Makefile
maya@maya:~/Desktop/OperatingSystems_Ex1/Question_3$ make all
gcc -Wall -g -c -fPIC libCodec.c
gcc -shared -o libCodec.so libCodec.o
gcc -Wall -g -c mainEncode.c
gcc -Wall -g -o encode mainEncode.o -L. -lCodec
gcc -Wall -g -c mainDecode.c
gcc -Wall -g -o decode mainDecode.o -L. -lCodec
```

```
OperatingSystems_Ex1 > Question_3 > Output.txt
1 234567890a ifmmp nz obnf jt Jtsbfm

Input.txt
1 1234567890 hello my name is Israel

maya@maya: ~/Desktop/OperatingSystems_Ex1/Question_3
maya@maya:~/Desktop/OperatingSystems_Ex1/Question_3$ make clean
rm -f libCodec.o mainEncode.o mainDecode.o encode decode libCodec.so
maya@maya:~/Desktop/OperatingSystems_Ex1/Question_3$ make all
gcc -Wall -g -c -fPIC libCodec.c
gcc -shared -o libCodec.so libCodec.o
gcc -Wall -g -c mainEncode.c
gcc -Wall -g -o encode mainEncode.o -L. -lCodec
gcc -Wall -g -c mainDecode.c
gcc -Wall -g -o decode mainDecode.o -L. -lCodec
maya@maya:~/Desktop/OperatingSystems_Ex1/Question_3$ ls
decode  libCodec.c  libCodec.so  mainEncode.c  Output.txt
encode  libCodec.h  mainDecode.c  mainEncode.o
Makefile Input.txt  libCodec.o  mainDecode.o  Makefile
maya@maya:~/Desktop/OperatingSystems_Ex1/Question_3$ export LD_LIBRARY_PATH=.
maya@maya:~/Desktop/OperatingSystems_Ex1/Question_3$ ./encode Input.txt Output.txt
```

חלק ד : עבודה עם תהליכים

אנחנו מעוניינים לבנות דוחס-מצפין כמו zip בעזרת מספר תהליכים

- Tape Archive - tar(1) - מאחה ספריה ותת ספריות (כולל מספר קבצים) לקובץ אחד גדול. בדומה יודע לפתוח קובץ גדול ומאחה למספר קבצים
- compress(1)/uncompress(1 או gzip(1)/gunzip(1) - דוחס קובץ גדול לקובץ קטן (דחוס) ופותח אותו (ניתן להשתמש בכל דוחס)
- gpg(1) - מצפין (או מפענח) קובץ

אנחנו מעוניינים לקבל שתי פקודות

myzip , myunzip

שתי הפקודות מקבלות פרמטר בעזרת argv.

במקרה של myzip זוהי ספריה. אנחנו מעוניינים לאחות אותה לקובץ גדול (בעזרת tar) לדחוס אותה (למשל

בעזרת compress) ולהצפין אותו בעזרת gpg.

במקרה של unzip זהו קובץ דחוס. אנחנו מעוניינים לפענח אותו ולפרוס אותו.

הנחיות-הקלות.

- תוכלו להניח את קיומם של מחיצת gpg עם מפתחות. אינכם נדרשים לייצר אותה בשום שלב.

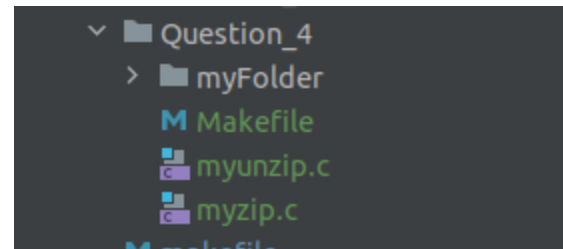
- תוכלו להניח שכל התהליכים (tar, compress, gpg) מצליחים. לא צריך לטפל במקרי קצה שהם לא מצליחים

- תוכלו להניח שהקלט חוקי (כלומר תמיד מפעילים אתכם עם פרמטר אחד והקובץ או הספריה שהם מתארים קיים)

- ניתן בהחלט ליצור קבצים זמניים (לדוגמא - נבצע tar וניצור קובץ זמני, נדחוס אותו, ונקבץ אותו) מותר להשתמש

בmktemp(1 או mktemp(3 ליצירת הקובץ הזמני.

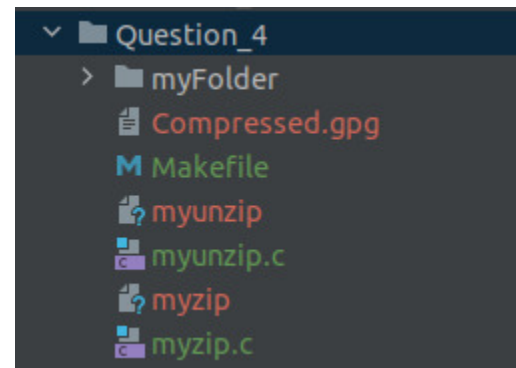
נבחין בתכולת התיקייה שלנו, אשר מכילה את קבצי ה C, ועוד תיקייה בשם myFolder המכילה כמה קבצים מסוגים שונים



כעת נקמפל את myzip ונריץ אותו עם פרמטר של שם הספרייה

```
ilan@ilan-Latitude-5420:~/CLionProjects/untitled1/Ex_1_OperatingSystems/Question_4$ make all
gcc -Wall -g myzip.c -o myzip
gcc -Wall -g myunzip.c -o myunzip
ilan@ilan-Latitude-5420:~/CLionProjects/untitled1/Ex_1_OperatingSystems/Question_4$ ./myzip myFolder 1234
Compression Completed.
ilan@ilan-Latitude-5420:~/CLionProjects/untitled1/Ex_1_OperatingSystems/Question_4$
```

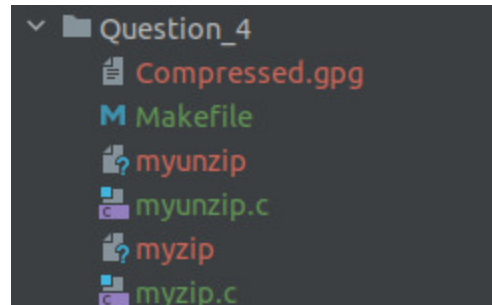
קיבלנו הודעה כי התהליך הושלם בהצלחה, כעת נראה כי אכן נוצר קובץ Compressed.gpg המכיל את הצופן



ואכן נוצר הקובץ כפי שנראה בתמונה וכי שנראה אותו מבפנים

[illegible]

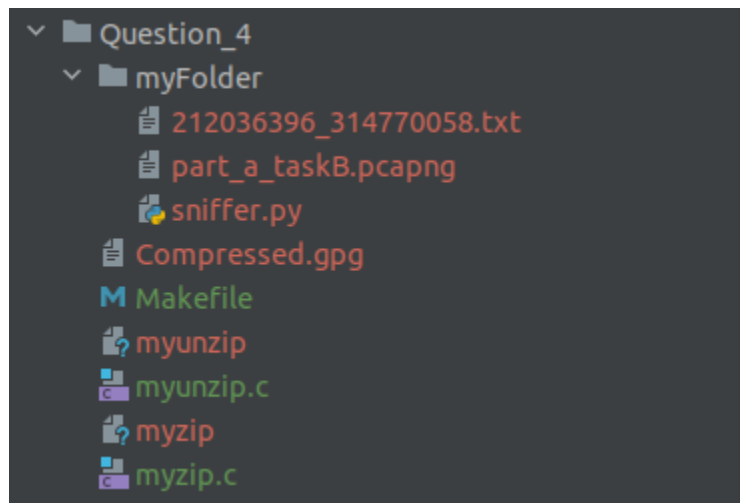
כעת, נמחק את התיקייה myFolder



לאחר שמחקנו אותה, נריץ את myunzip עם פרמטר לקובץ Compressed.gpg ונראה כיצד הוא משחזר את התיקיה בצורתה המקורית

```
ilan@ilan-Latitude-5420:~/CLionProjects/untitled1/Ex_1_OperatingSystems/Question_4$ ./myunzip Compressed.gpg 1234
gpg: AES256.CFB encrypted data
gpg: encrypted with 1 passphrase
myFolder/
myFolder/212036396_314770058.txt
myFolder/part_a_taskB.pcapng
myFolder/sniffer.py
ilan@ilan-Latitude-5420:~/CLionProjects/untitled1/Ex_1_OperatingSystems/Question_4$
```

כפי שניתן לראות, התיקיה שוחזרה כולל הקבצים הפנימיים במלואה



הגישו

1. את הקוד של כל חלק (מתוך ה 4) בתיקיה נפרדת, הכוללת את קבצי הקוד, וקובץ makefile **תקין**

2. קובץ makefile רקורסיבי שבונה את כולן שנימצה מחוץ לתיקיות
3. בהחלט ניתן לצרף קובץ readme וצילומי מסך (לא מהפלאפון)
4. את כל התוצרים "אורזים" בקובץ ZIP, ששמו הוא ה ת.ז של המגיש. במידה ומגישים שנים, שם הקובץ יהיה תז_תז. **קבצים שהם לא ZIP לא יבדקו**

הערה : אם עובדים בזוגות מומלץ לעבוד בזוג ולא לחלק את המטלה - המטלה מנסה לתת לכם כישורים בסיסים שידרשו בקורס ואם תחלקו אותה ל 2 לאחד הסטודנטים הכישורים יהיו חסרים.

איחור בהגשת המטלה ללא סיבה מוצדקת (מילואים/מחלה) מוריד 10 נקודות ליום. במידה ולאחד משני השוטפים יש סיבה מוצדקת, ולשני לא, ההורדה תיה 5 נקודות ליום.

יש להיזהר בזמן העברת הקבצים בין מערכות הפעלה שונות (וחלונות במיוחד), יש לוודא שלקבצים שמות תקינים. ללא .txt , גרשיים, או כל ארטיפקט אחר. במידה והוגשו הקבצים בשמות שגויים, ההורדה היא 20 נקודות.