Handout

Series S03:
- Control Flow
- gdb & LLDB
- AST & CS

Prologue

You will have to test your solutions of Exercises 1 and 2 on machine: for your convenience a template is available on Moodle. A working program for exercise 4 is available at the same place.

Reading

a) Browse/Read [KR88], chap. 3 : “Control Flow”

b) Study the lecture notes Study the tutorial "AST and CS”

c) Study the tutorial "gdb" [TU03] (examples I and II).
Hint: don't worry if you not yet understand the pointer operator '' of the function swap2() in Example I. Just keep it. We will come back later.*

1. Loops: 'while', 'for' and 'do while'

a) Replace the `while` loop of Fig. 1a by a `for` loop. Test your solution on machine.

```
low = 0;
high = n - 1;
while (low <= high) {
    // do some actions
    ++low;
}
```

Fig. 1a

b) Same exercise as (a), but use a `do-while` loop this time. Hint: beware if `n <= 0`. Test your solution on machine.

UNI FR

2. goto & label, switch, break and continue

a) Replace the `for` loop of Fig. 2a by a `goto` statement. Test your solution on machine.

```
for (i = 0; i < n; i++) {
 // do some actions
}
```

Fig. 2a

b) Rewrite the switch statement of Fig. 2b with gotos and labels. Test your solution on machine.

```
switch (i) {
  case 1: printf("case 1\n"); break;
  case 2: printf("case 2\n");  // Beware: no break !!!
 default: printf("default case\n"); break;
}
```

Fig. 2b

c) Replace the `break` of Fig. 2c by a `goto`. Test your solution on machine.

```
for (i = 0; i < n; i++) {
  // do some actions 1
  if (<cond1>) break;
  // do some actions 2
}
```

Fib. 2c

d) Replace the `continue` of Fig. 2d by a `goto`. Test your solution on machine.

```
for (i = 0; i < n; i++) {
  // do some actions 1
  if (<cond1>) continue;
  // do some actions 2
}
```

Fig. 2D

3. gdb – 1st visit

Compile and execute the program of figure 4 with `gdb`. Then conceive and comment a pertinent **debug script** (i.e. a sequence of gdb commands) for it, similar to the ones of Examples 1 and 2 of the "gdb tutorial" [TU03].

Hint: observe the values of 'ctr' and 'i' at the end of lines 4 and 7; and add a watch on the value of 'i' and 'res' while running the loop.

```
#include <stdio.h>
 int N = 3;
 int main() {
   int ctr, i;  // line 4
   int res;

   i = N;       // line 7
   res = N;

   printf("res  N  i\n");
   for (ctr = 0; ctr < N; ++ctr, --i) {
      res = N/i;
      printf("%3i%3i%3i\n",res, N, i);
   }

   return 0;
 }
```

4. gdb – 2nd visit

Same exercise as (4) but now replace at line 11 the operator < by <=. Let the program crash and observe the core dump file with gdb.

Hint:

- The core dump will be generated when you run the compiled program from the terminal. If no core dump file was generated, try running the following command then compile again:
  `% ulimit -c unlimited`
- The command required to read the dump file with GDB is:
  `% gdb <binary_file_path> <core_file_path>`
- Use the following commands to investigate the error scenario captured in the core dump:
  ```
  (gdb) bt
   (gdb) list
   (gdb) info locals
   (gdb) print <expression>
  ```
- After checking the core dump with GDB, indicate which line caused the error and identify what error signal was issued by the OS. After identifying the signal, you can find out what it means on this page:
  http://www.gnu.org/software/libc/manual/html_node/Program-Error-Signals.html

5. Abstract Syntax Tree (AST) and Control Stack (CS) for an expression

Draw the AST and CS for significant steps (1) for the statement: '`n=2*m+n;`' and (2) for the expression: '`n*=m+1`', where `n=6` and `m=3`.

Hint: in these 2 simple cases draw all steps representing an elementary action as reading an item (variable, number, operator, semi-colon).

Hand in.

Upload your answers on Moodle.

References

[KR88] B. Kernighan, D. Ritchie, The C Programming Language, 2nd Ed., Prentice Hall, 1988.

[TU03] Moodle > Tutorial "TU03 gdb"

[TU03] If you like to work on macOS directly, have a look at the document on Moodle > Tutorial "TU03 gdb on macOS". (We recommend to use a Virtualbox with Ubuntu for this course.)