COMsW 1003-1

Introduction to Computer Programming in C

Lecture 8

Spring 2011

Instructor: Michele Merler

http://www1.cs.columbia.edu/~mmerler/comsw1003-1.html

Announcements

Homework 1 correction out this afternoon

Homework 2 is out

- Due Monday, February 28th
- Start early (especially Exercise 2)!

Today

- Functions
- Recursion
- Debugging (if time)

Infinite Loops

- Loops where the condition is always TRUE

- Will stop only with:
  - `break`
  - modification of the condition variables

```
while ( 1 ){

  /* body modifies x */

  if( x!= 0 ) {
        break;
  }

}
```

Infinite Loops

- Loops where the condition is always TRUE

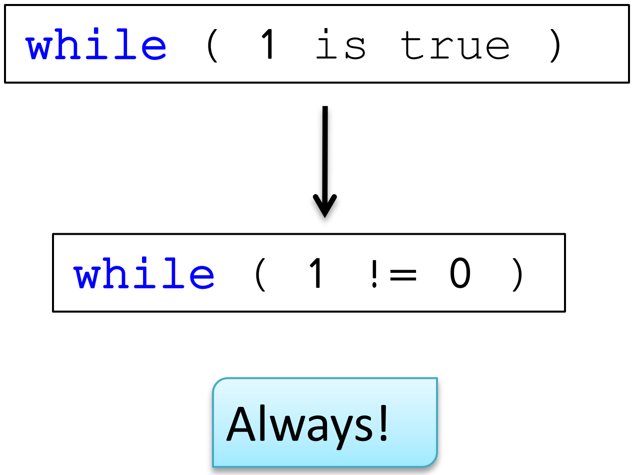
- Will stop only with:
  - `break`
  - modification of the condition variables

```
while ( 1 ){

   /* body modifies x */

   if( x!= 0 ) {
        break;
   }

}
```



```
while ( 1 is true )
```

```
while ( 1 != 0 )
```

Always!

Operators - Logical

- A variable with value 0 is false, a variable with value !=0 is true

```
int x = 3, y = 0, z, k, t, q = -3;

z = x && y;     // z = 0;      x is true but y is false

k = x || y;     // k = 1;      x is true

t = !q;         // t = 0;      q is true
```

Infinite Loops

- Loops where the condition is always TRUE

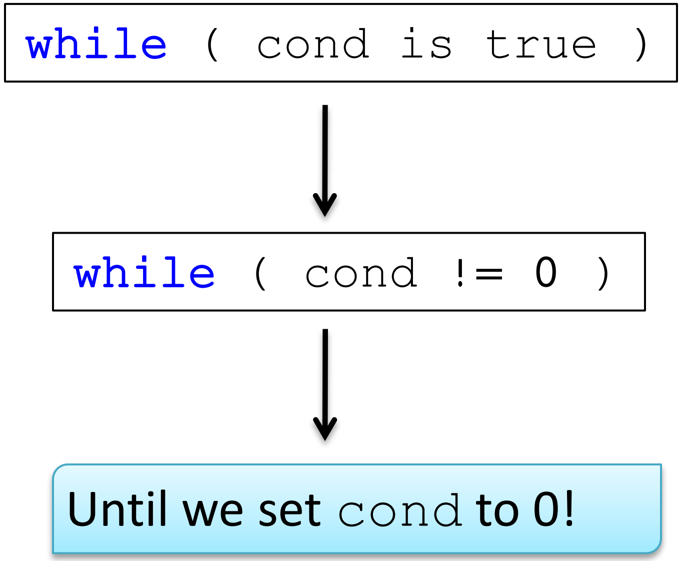
- Will stop only with:
    - `break`
    - modification of the condition variables

```
int cond = 7;

while ( cond ){

   /* body */
   if( x[3][5] != 7 ){
      cond = 0;
   }

}
```



`while ( cond is true )`

`while ( cond != 0 )`

Until we set `cond` to 0!

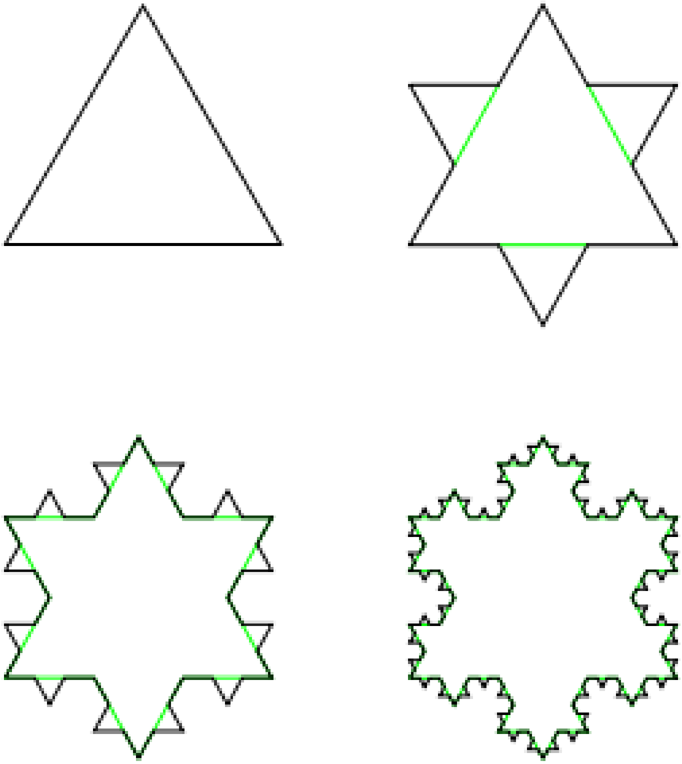
calculator.c

Functions Example

- Simple calculator
- Program that computes one basic arithmetic operation between 2 numbers

Functions - Recursion

- What if a function calls itself? Recursion



Functions - Recursion

- A recursive function must have two properties:
  - **Ending point** (i.e. a terminating condition)
  - **Simplify the problem** (every call is to a simpler input)

Example: Fibonacci sequence

In mathematics, famous numbers following the sequence

0 1 1 2 3 5 8 13 21 34 55 89 …

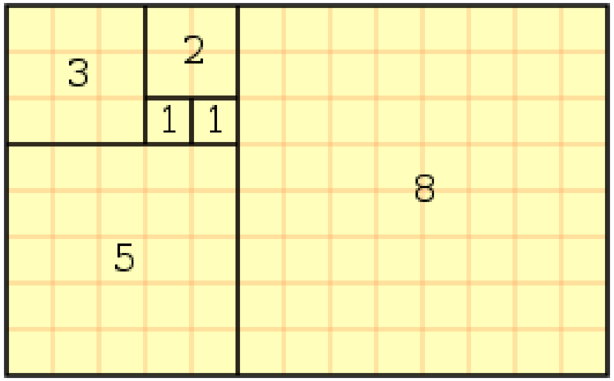
Given $F_0 = 0$, $F_1 = 1$ can be computed with recurrence $\boxed{F_n = F_{n-1} + F_{n-2}}$

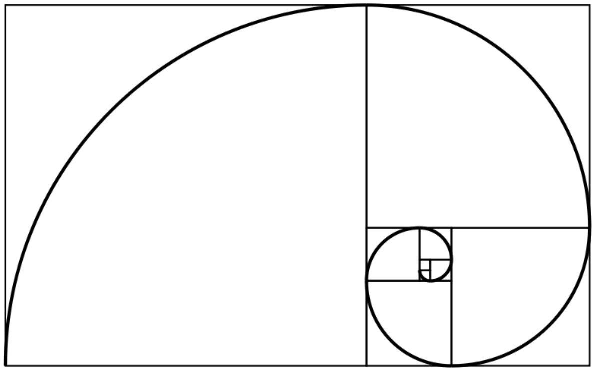
Code to compute the first 100 Fibonacci numbers:

```
int i = 0;
int fib[100];

fib[0] = 0;
fib[1] = 1;

for( i = 2; i < 100 ; i++ ) {
   fib[i] = fib[i-1] + fib[i-2];
}
```





Functions - Recursion

recursiveFib.c

- What if a function calls itself? Recursion
- What is the value of the number at position num in the Fibonacci sequence?

```
/* Fibonacci value of a given position in the sequence */
int fib ( int num ) {

  switch(num) {
   case 0:
       return(0);

   case 1:
       return(1);

   default: /* Including recursive calls */
       return(fib(num - 1) + fib(num - 2));

  }
}
```

Why are there no breaks ?

Functions - Recursion

recursiveFib.c

- What if a function calls itself? Recursion
- What is the value of the number at position num in the Fibonacci sequence?

```c
/* Fibonacci value of a given position in the sequence */
int fib ( int num ) {

  switch(num) {
   case 0:
        return(0);

   case 1:
        return(1);

   default: /* Including recursive calls */
       return(fib(num - 1) + fib(num - 2));

  }
}
```

Ending Points

Simplify problem

Debugging

Debugging

- Debugging consists basically in finding and correcting **run-time errors** in your program

- Multiple ways of doing it
  - Manual runs (for small programs)
  - Insert `printf()` in key lines

- There also exist INTERACTIVE debugging tools

- We will now see a basic one for UNIX: **gdb**

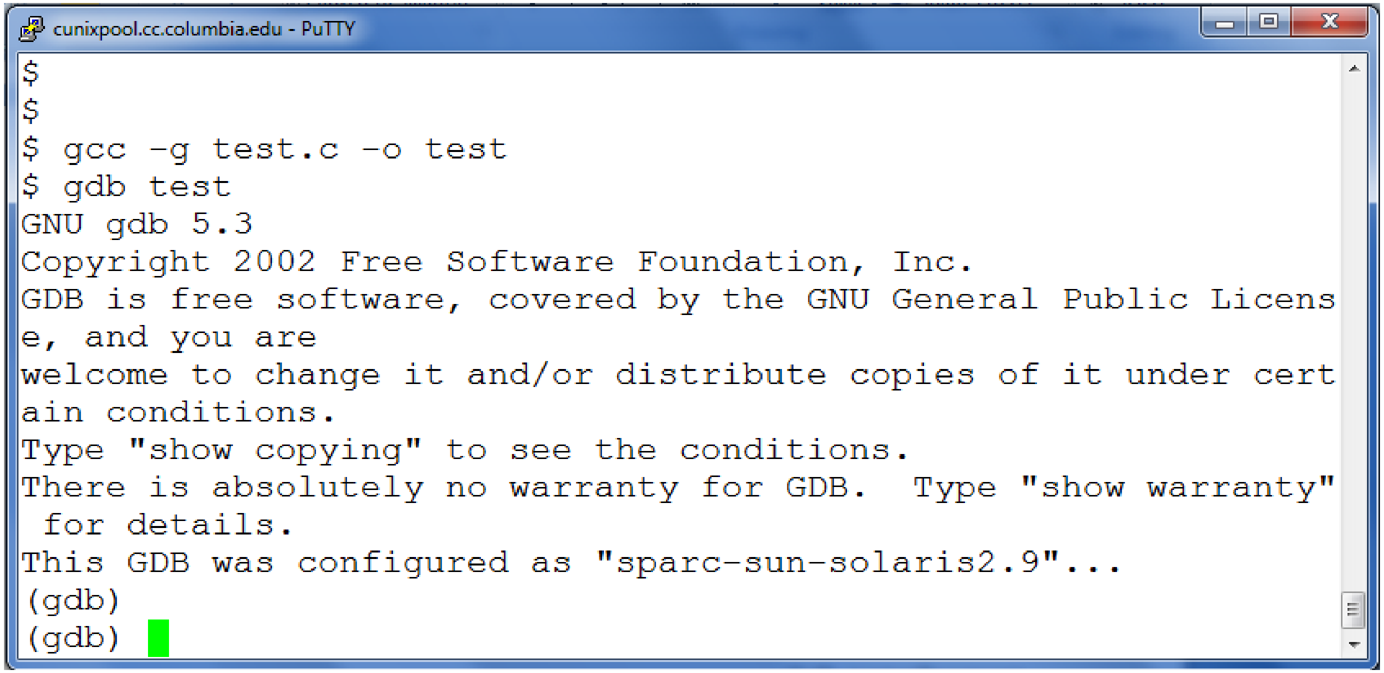
gdb

1. In order to use gdb on a program, we must use the **–g** option when compiling it

```
gcc  -g program.c   -Wall -o nameOfExecutable
```

2. Then, we can use the gdb command to start the interactive debugging environment

```
gdb nameOfExecutable
```



```
$
$
1. $ gcc -g test.c -o test
2. $ gdb test
GNU gdb 5.3
Copyright 2002 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public Licens
e, and you are
welcome to change it and/or distribute copies of it under cert
ain conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB.  Type "show warranty"
 for details.
This GDB was configured as "sparc-sun-solaris2.9"...
(gdb)
(gdb)
```

gdb commands

- **run** : run executable (program)currently watched.

```
(gdb) run
```

- **kill** : kill current execution of program

```
(gdb) kill
```

- **list** : show program source code

`(gdb) list 2,8` : shows lines 2 to 8 from source program

- **print** : print value of a variable or expression at the current point

```
(gdb) print buf
```

gdb commands

- **break** : insert breakpoint in program. Debugging run will stop at the breakpoint

```
(gdb) break  nameSource.c : lineNumber
(gdb) break  test.c: 12
```

- **next** : step to the next line (execute current line)

```
(gdb) next
```

- **continue** : continue with execution until next breakpoint or end of program

```
(gdb) continue
```

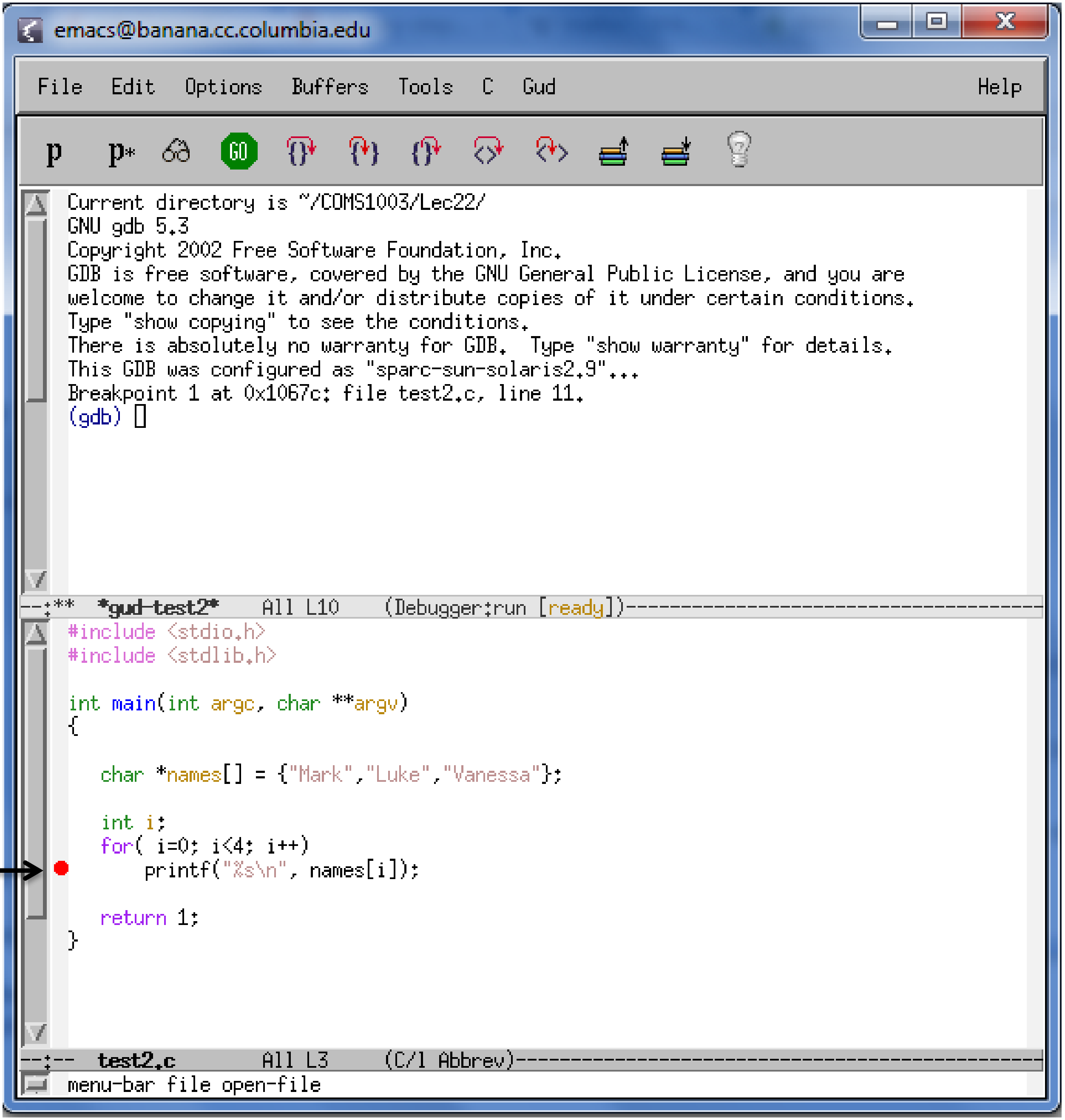
- **Quit** : exit gdb

```
(gdb) quit
```

Graphical GDB

- gdb can be run from Emacs

- Press `M-x` (in Windows `Esc-x`)
- Insert `gdb`
- Insert `executableName`
- Visual debugger


emacs@banana.cc.columbia.edu
File Edit Options Buffers Tools C Gud Help
p p* GO

```
Current directory is ~/COMS1003/Lec22/
GNU gdb 5.3
Copyright 2002 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public License, and you are
welcome to change it and/or distribute copies of it under certain conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB.  Type "show warranty" for details.
This GDB was configured as "sparc-sun-solaris2.9"...
Breakpoint 1 at 0x1067c: file test2.c, line 11.
(gdb) 
```

```
--:**  *gud-test2*     All L10     (Debugger:run [ready])-----------------------
```

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

int main(int argc, char **argv)
{

   char *names[] = {"Mark","Luke","Vanessa"};

   int i;
   for( i=0; i<4; i++)
●      printf("%s\n", names[i]);

   return 1;
}
```

```
--:--  test2.c         All L3      (C/l Abbrev)---------------------------------
menu-bar file open-file
```

Can enable breakpoints with a click