PA2 tutorial
- from the point of getting code to work

Mingyuan Xia

COMP 310, Fall 2012

Tools

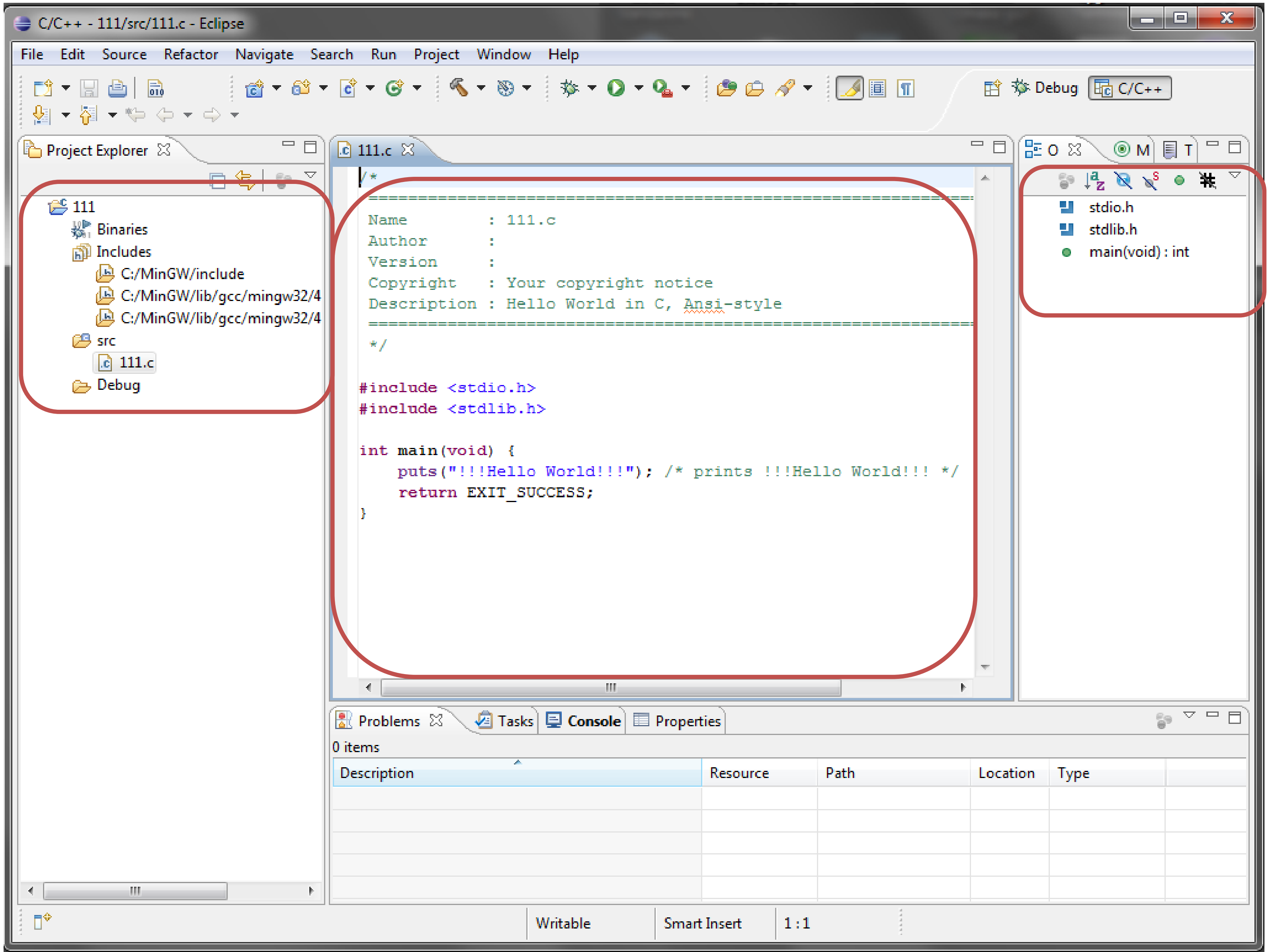
- IDE: eclipse + CDT
  - Check compile errors
  - Debug
- Valgrind
  - Check memory related problems
  - Segfault! Glibc free crash!

Eclipse

- Windows, Mac, Linux
- IDE for Java, C++, Python…

http://www.eclipse.org/downloads/packages/eclipse-ide-cc-developers/galileosr2

Eclipse

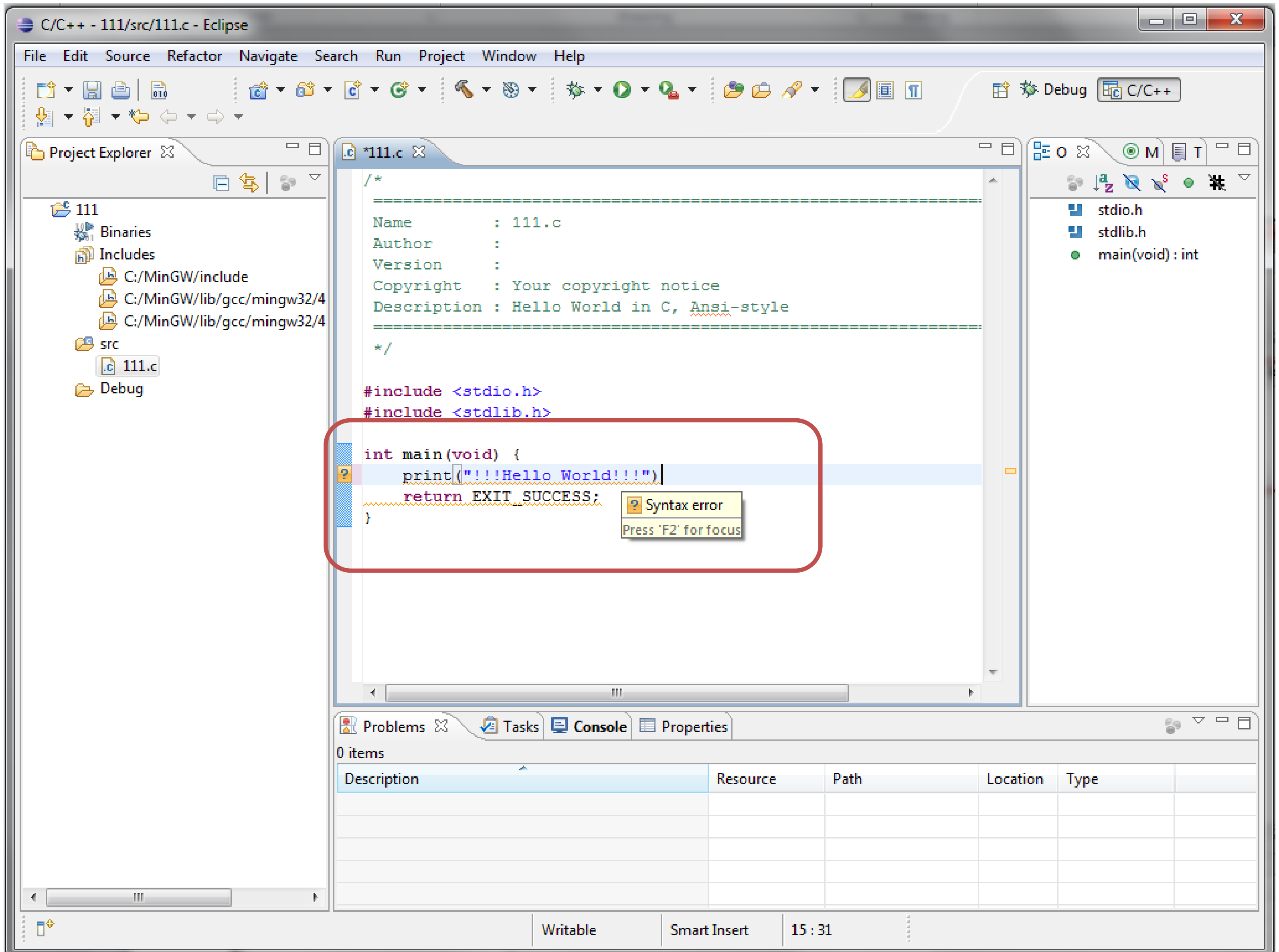


```c
/*
 ============================================================================
 Name        : 111.c
 Author      :
 Version     :
 Copyright   : Your copyright notice
 Description : Hello World in C, Ansi-style
 ============================================================================
 */

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

int main(void) {
	puts("!!!Hello World!!!"); /* prints !!!Hello World!!! */
	return EXIT_SUCCESS;
}
```

Compile errors



C/C++ - 111/src/111.c - Eclipse

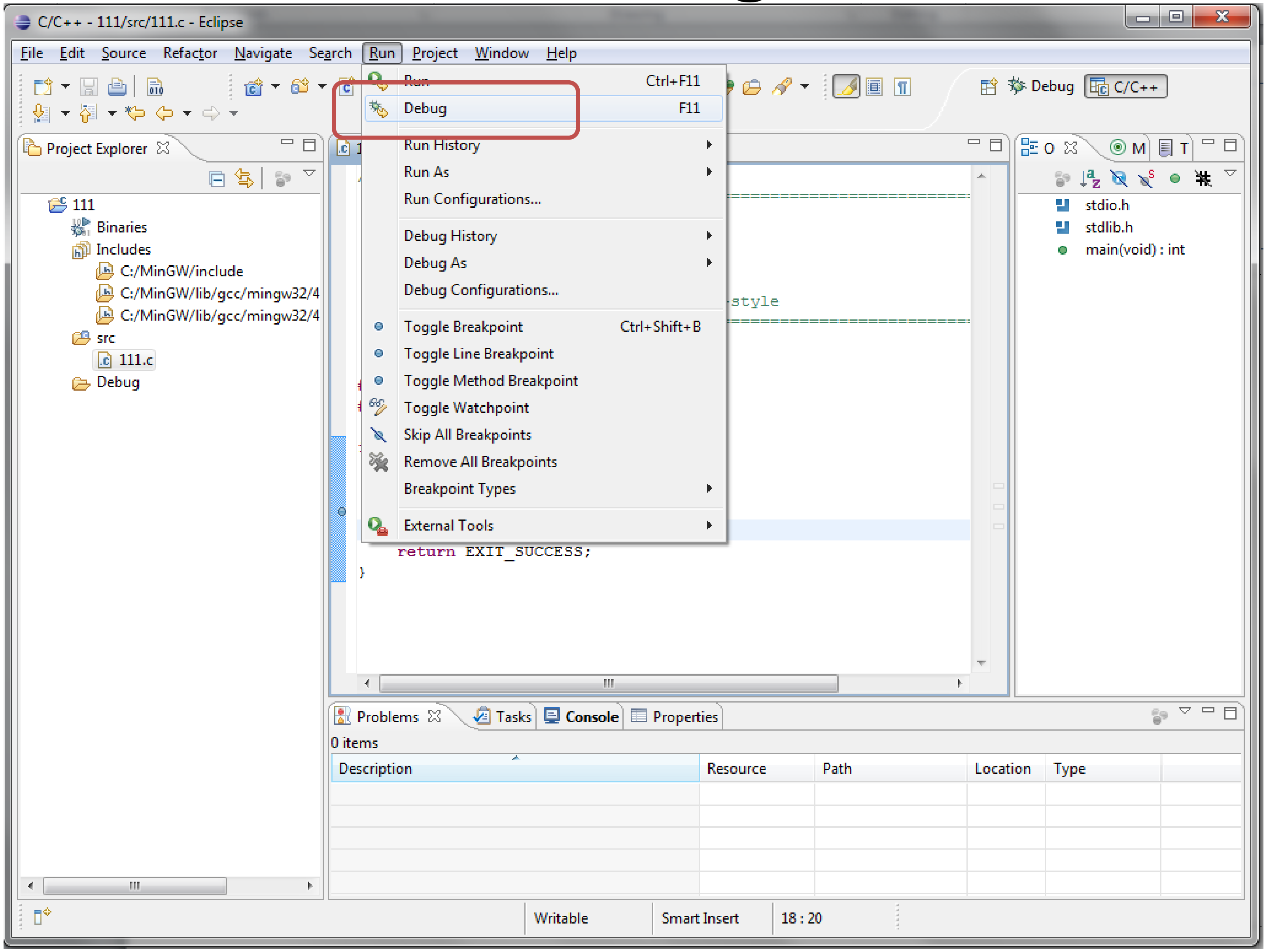
```
/*
 ============================================================================
 Name        : 111.c
 Author      :
 Version     :
 Copyright   : Your copyright notice
 Description : Hello World in C, Ansi-style
 ============================================================================
 */

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

int main(void) {
    print("!!!Hello World!!!")
    return EXIT_SUCCESS;
}
```

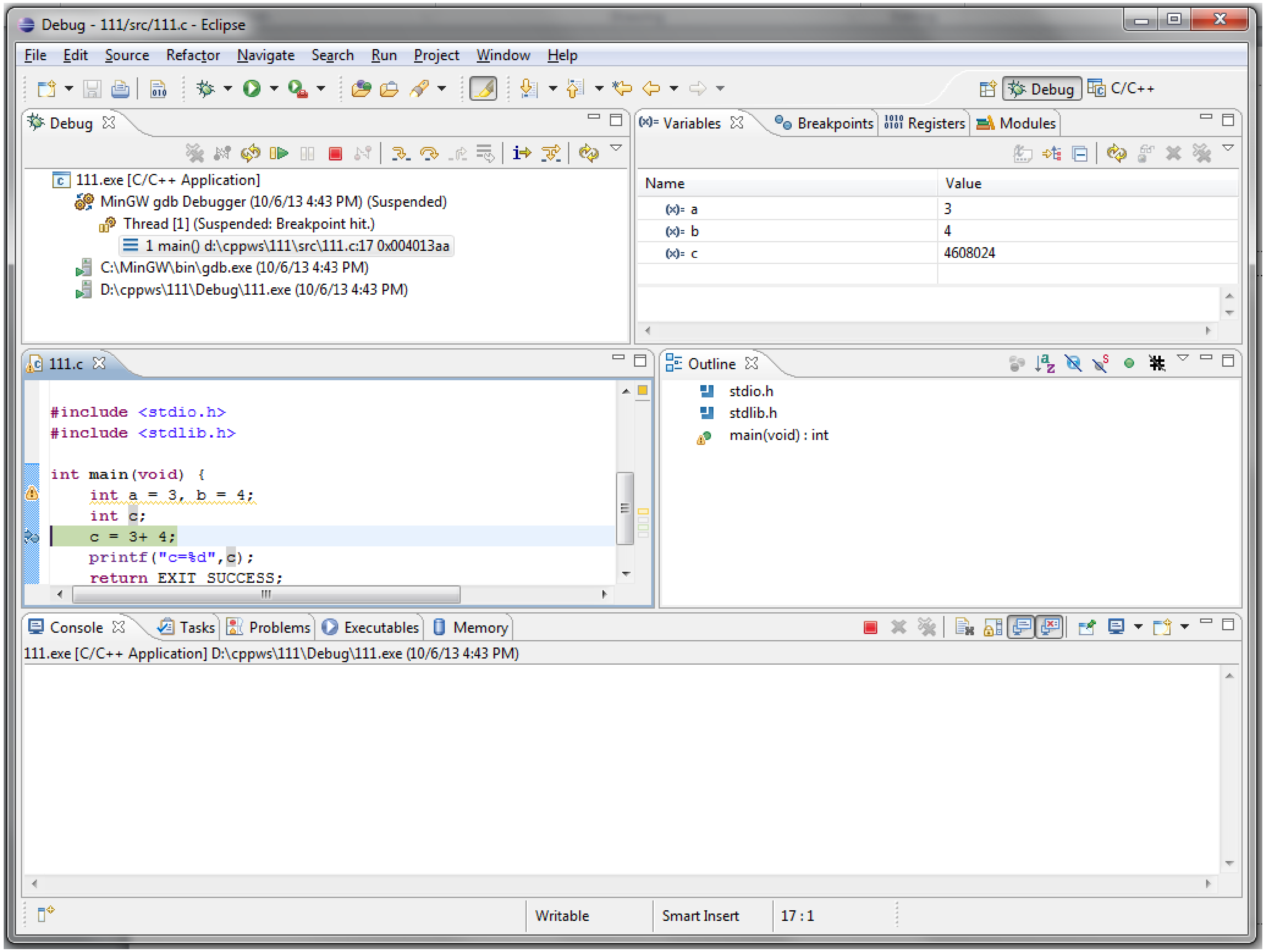
Syntax error
Press 'F2' for focus

Debug



```
    return EXIT_SUCCESS;
}
```

Debug



```
111.exe [C/C++ Application]
    MinGW gdb Debugger (10/6/13 4:43 PM) (Suspended)
        Thread [1] (Suspended: Breakpoint hit.)
            1 main() d:\cppws\111\src\111.c:17 0x004013aa
    C:\MinGW\bin\gdb.exe (10/6/13 4:43 PM)
    D:\cppws\111\Debug\111.exe (10/6/13 4:43 PM)
```

Name	Value
a	3
b	4
c	4608024

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

int main(void) {
    int a = 3, b = 4;
    int c;
    c = 3+ 4;
    printf("c=%d",c);
    return EXIT_SUCCESS;
```

```
111.exe [C/C++ Application] D:\cppws\111\Debug\111.exe (10/6/13 4:43 PM)
```

Valgrind

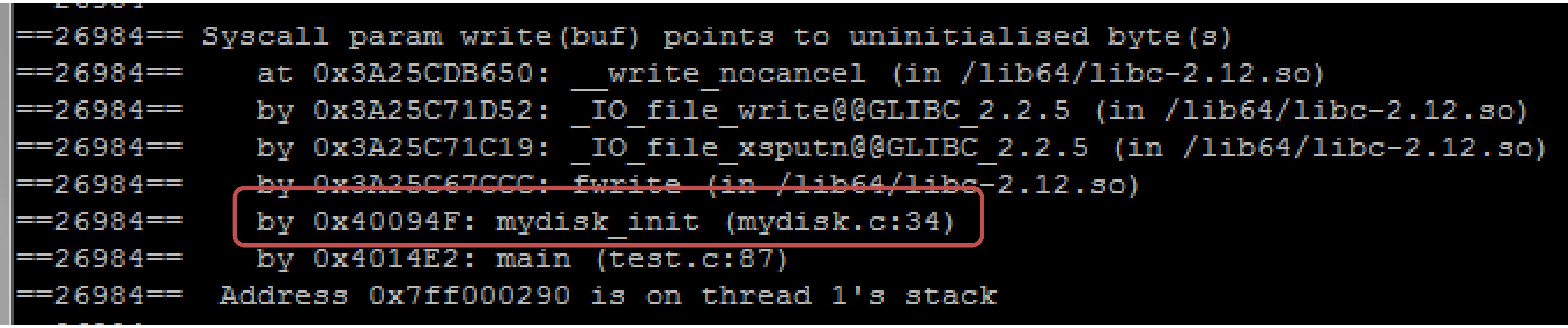
- Memory problems (#1 C problem)
  - Segmentation fault (35% occurrence in PA1)
  - Glibc complains on calling free() (10% occurrence in PA1)
  - Hidden -_______-

- Valgrind runs with your program and checks memory-related problems
- Linux*

http://valgrind.org/

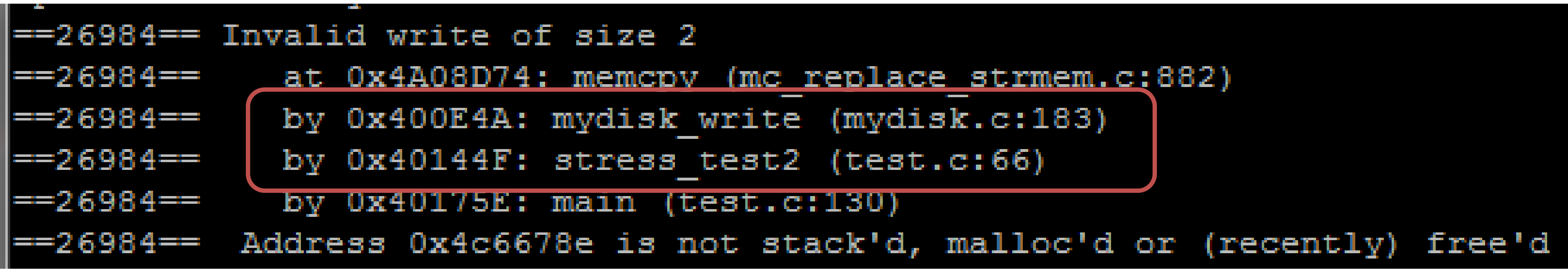
Valgrind: howto

- Command line: valgrind ./mydisk



```
==26984== Syscall param write(buf) points to uninitialised byte(s)
==26984==    at 0x3A25CDB650: __write_nocancel (in /lib64/libc-2.12.so)
==26984==    by 0x3A25C71D52: _IO_file_write@@GLIBC_2.2.5 (in /lib64/libc-2.12.so)
==26984==    by 0x3A25C71C19: _IO_file_xsputn@@GLIBC_2.2.5 (in /lib64/libc-2.12.so)
==26984==    by 0x3A25C67CCC: fwrite (in /lib64/libc-2.12.so)
==26984==    by 0x40094F: mydisk_init (mydisk.c:34)
==26984==    by 0x4014E2: main (test.c:87)
==26984==  Address 0x7ff000290 is on thread 1's stack
```

Valgrind: howto



```
==26984== Invalid write of size 2
==26984==    at 0x4A08D74: memcpy (mc_replace_strmem.c:882)
==26984==    by 0x400E4A: mydisk_write (mydisk.c:183)
==26984==    by 0x40144F: stress_test2 (test.c:66)
==26984==    by 0x40175E: main (test.c:130)
==26984==  Address 0x4c6678e is not stack'd, malloc'd or (recently) free'd
```

Valgrind: howto

- Bugs
  - Write to a freed location
  - Read an uninitialized location
- How to fix?
  - Debug the program with eclipse
  - Set a breakpoint at the problematic line

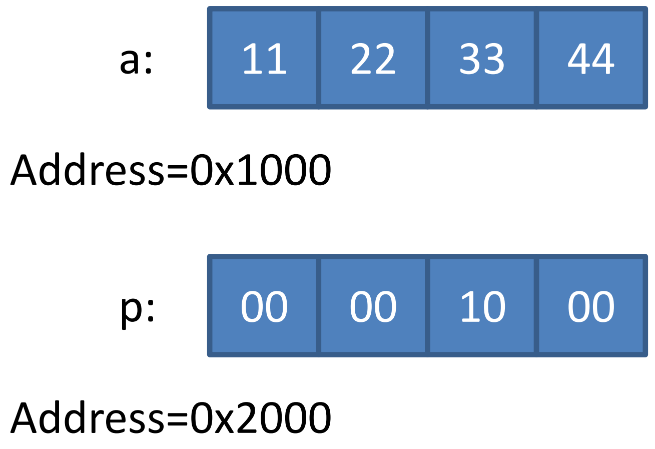
Understand the language

- Pointers, variables and arrays
- Casting and reinterpreting memory
- sizeof, strlen, strcpy
- Bit operations

Variables and pointers

Definition

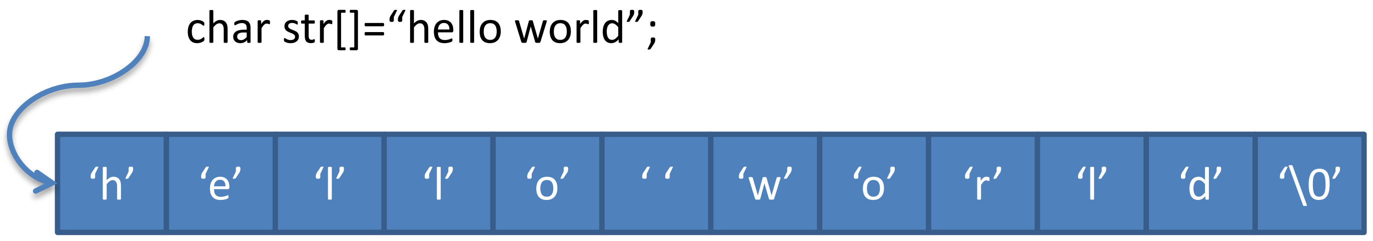
```
int a=0x11223344;
int *p = &a;
```



Use

	value	type
a	0x11223344	Int
&a	0x1000	Int*
p	0x1000	Int*
*p	0x11223344	int
&p	0x2000	Int **

Arrays and pointers



Address=0x1000

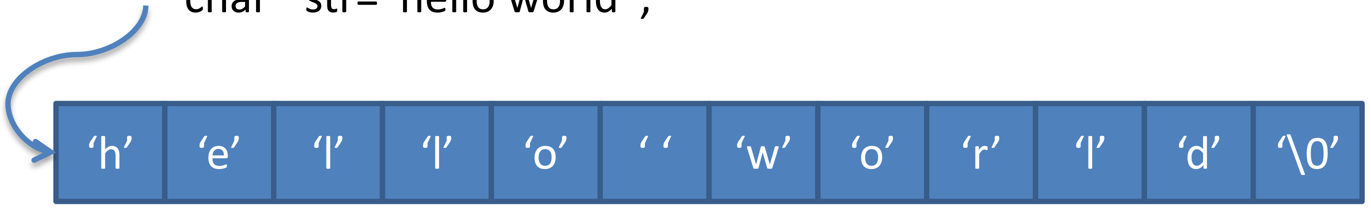
Use

	value	type
str[0]	'h'	char
str	0x1000	char*

An array is a pointer

Moving inside an array

char *str="hello world";

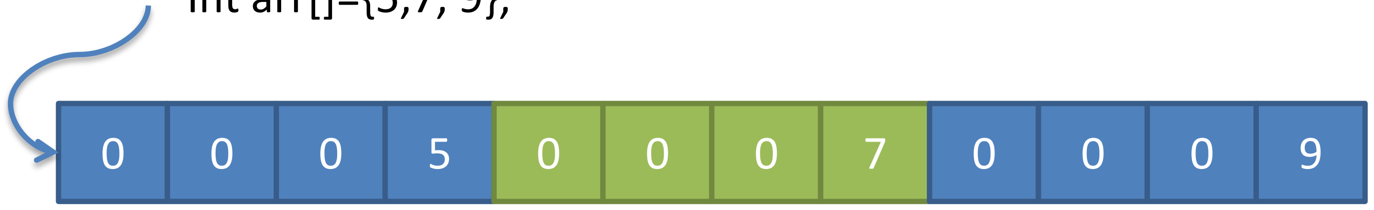


Address=0x1000

	value	type
str[0]	'h'	char
str[1]	'e'	char
str+1	0x1001	Char*
*(str+1)	'e'	char
(str+1)[1]	'l'	char

Arrays (cont.)

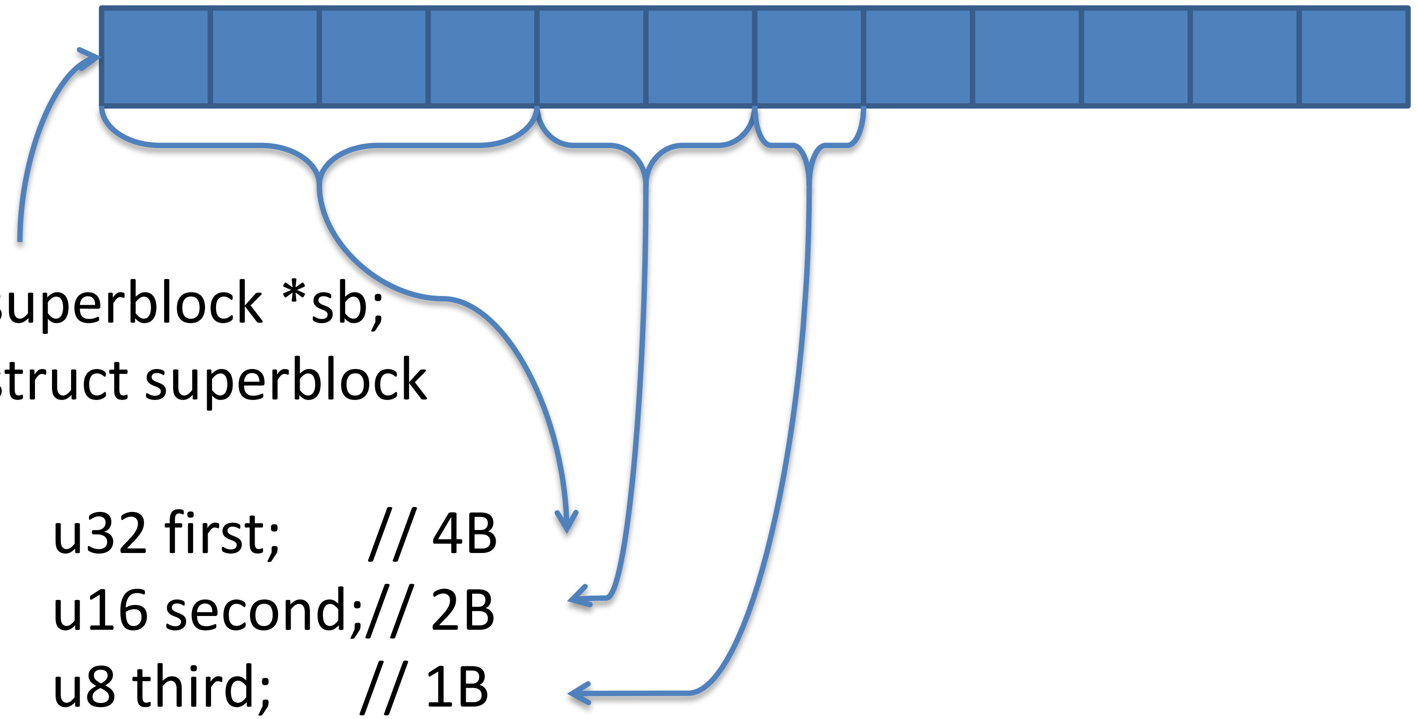
Int arr[]={5,7, 9};



Address=0x1000

	value	type
arr[0]	5	int
arr	0x1000	int*
arr+1	**0x1004**	Int *
*(arr+1)==arr[1]	7	Int
(arr+1)[1]==arr[2]	9	int

Struct



```
superblock *sb;
struct superblock
{
    u32 first;      // 4B
    u16 second;// 2B
    u8 third;      // 1B
};
```

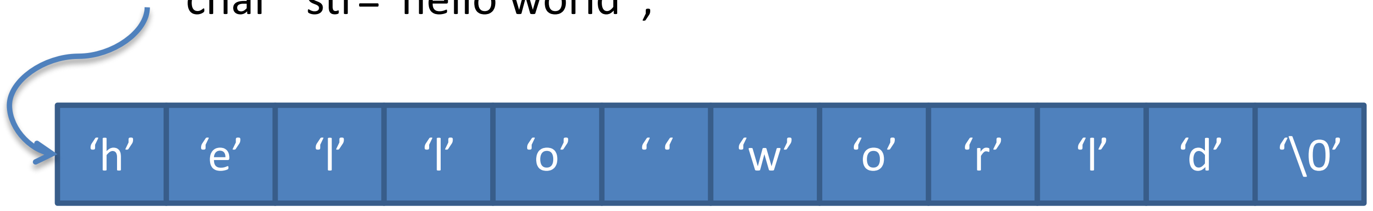
Use pointers correctly

- A pointer can points to one variable or an array …
  - one variable is an array of only one element
- Always know where your pointer points to
  - What is your pointers’ limit
  - E.g. p points to single variable, then *p, p[0] are valid while p[1] is not

casting

- Reinterpret the memory

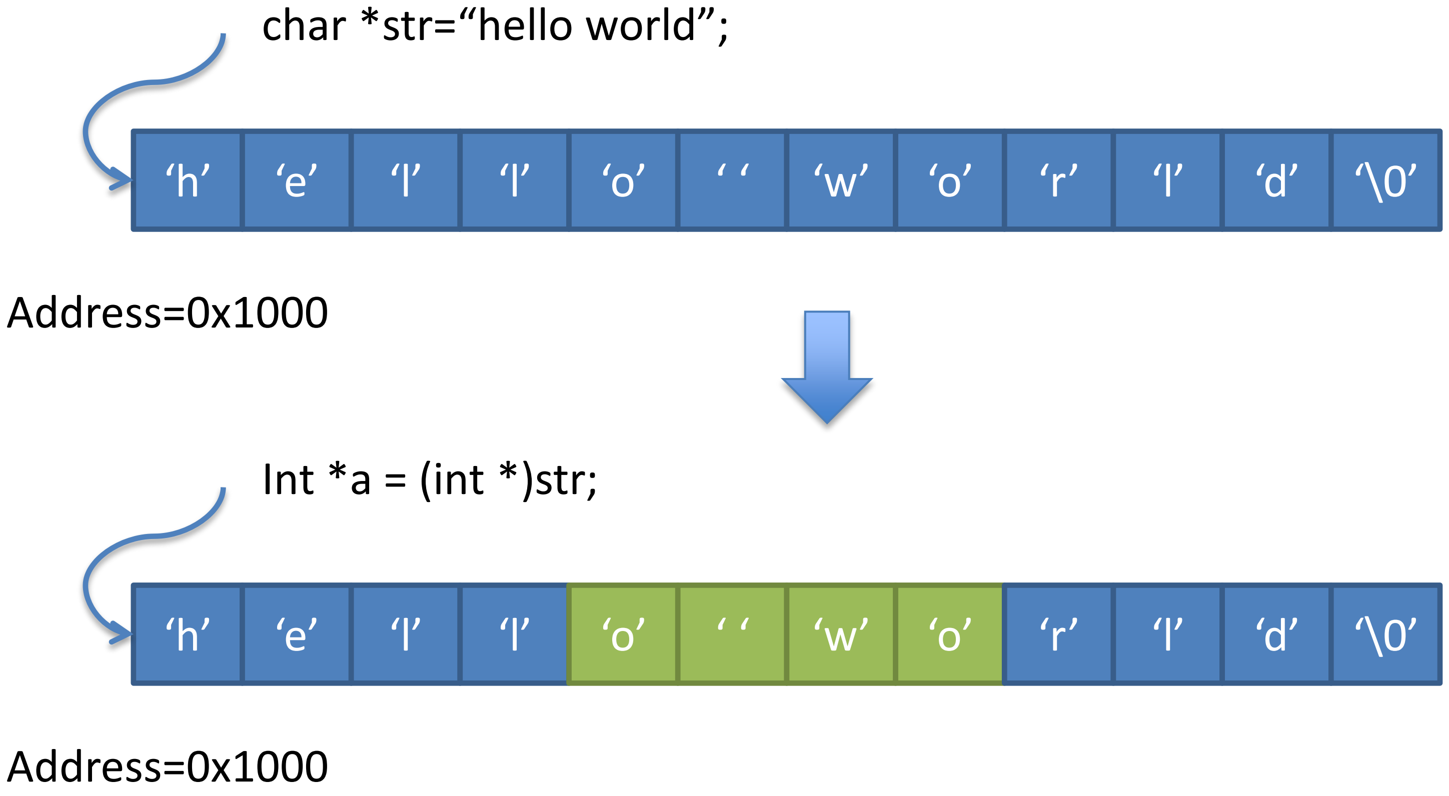
char *str=“hello world”;



Address=0x1000

casting

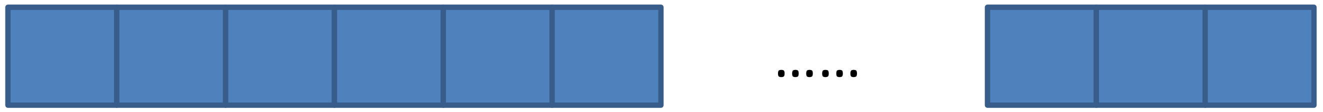
- Reinterpret the memory

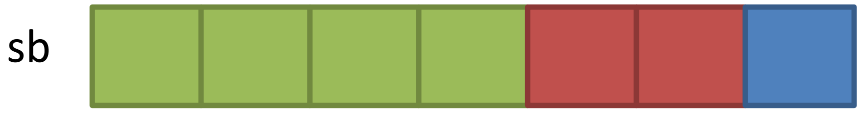

char *str=“hello world”;
‘h’ ‘e’ ‘l’ ‘l’ ‘o’ ‘ ‘ ‘w’ ‘o’ ‘r’ ‘l’ ‘d’ ‘\0’
Address=0x1000
Int *a = (int *)str;
‘h’ ‘e’ ‘l’ ‘l’ ‘o’ ‘ ‘ ‘w’ ‘o’ ‘r’ ‘l’ ‘d’ ‘\0’
Address=0x1000

casting

- Block to superblock, inode, directory,

Char buffer[BLOCK_SIZE]





```
struct superblock
{
    u32 first;     // 4B
    u16 second;// 2B
    u8 third;      // 1B
};
```

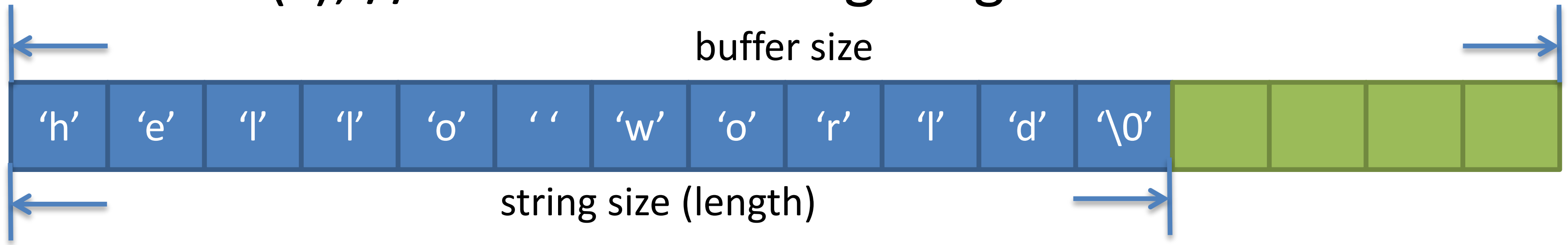
Size of “memory”

- buffer size, string size, sizeof

  char *s = (char *)malloc(16);

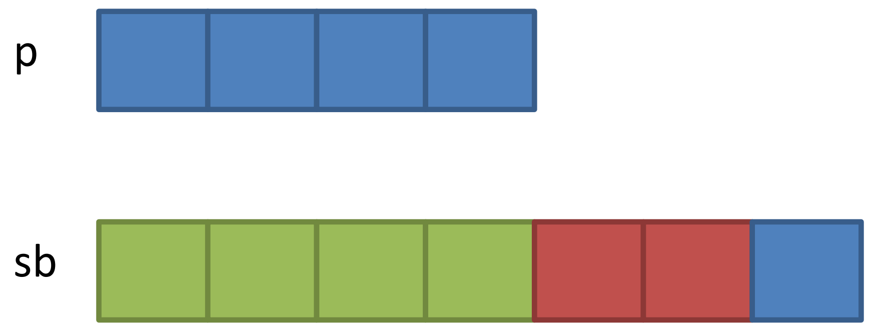
  strcpy(s, ”hello world!”);

  strlen(s); // return the string length



- sizeof(s) = sizeof(its type) = sizeof(char *) = 4

Sizeof



```
struct superblock
{
    u32 first;      // 4B
    u16 second;// 2B
    u8 third;       // 1B
};
```

- superblock sb;
- superblock *p = &sb;

- sizeof(p) = sizeof(superblock *) = 4
- sizeof(sb) = sizeof(superblock) = 4+2+1 = 7
- sizeof(*p) =?

PA2 hints

- read a “superblock” from the disk

```
sfs_read_block(char *buf, int bid);
char *b = malloc(BLOCK_SIZE);
sfs_read_block(b, x);
superblock *sb = (superblock *)b;
….
free(b); or free(sb);
// the free is able to tell the buffer size
```

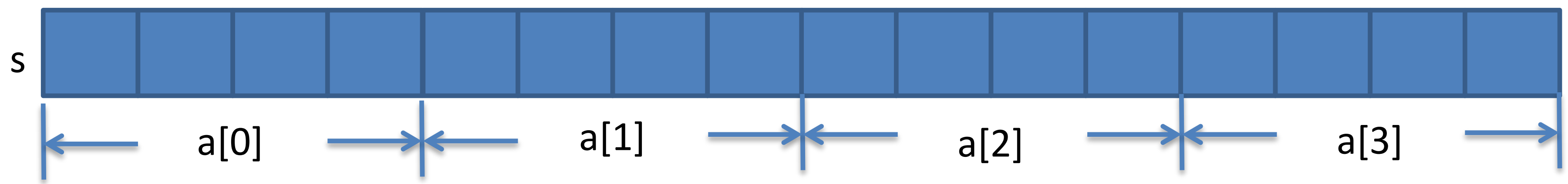
Even simpler

- Assume superblock occupies one block
  - i.e., sizeof(superblock) = BLOCK_SIZE

```
superblock sb; // not a pointer!
sfs_read_block((char*)&sb, x);
// use sb
// Do not need to free sb
```

You can also …

- char *s = (char *)malloc(16);
- u32 *a = (u32 *)s;

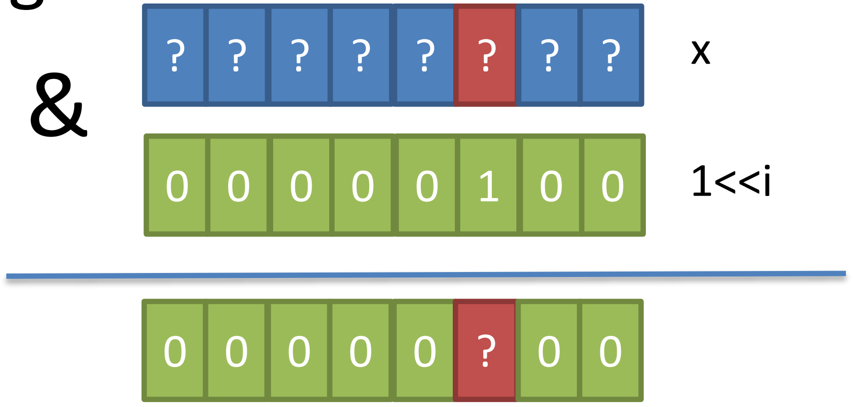


- You will use this for freemap management in PA2

Bit operation

- << (left shift), & (AND), | (OR), ^ (XOR), ~(NOT)

- Fetch the i-th bit of an integer x
  - x & (1<<i)
- Set the i-th bit of x
  - x| (1<<i)
- Unset the i-th bit of x
  - x & ~(1<<i)



- More: http://en.wikipedia.org/wiki/Bit_manipulation

- Thank you Q/A