

week 5, continued

pre-processing

compiling

assembling

linking

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

```
int main(void)
```

```
{
```

```
    printf("hello, world\n");
```

```
}
```

```

        .file      "hello1.c"
        .text
        .globl    main
        .align    16, 0x90
        .type     main,@function
main:                                       # @main
# BB#0:
    pushl   %ebp
    movl   %esp, %ebp
    subl  $24, %esp
    leal  .L.str, %eax
    movl  $0, -4(%ebp)
    movl  %eax, (%esp)
    calll printf
    movl  $0, %ecx
    movl  %eax, -8(%ebp)          # 4-byte Spill
    movl  %ecx, %eax
    addl  $24, %esp
    popl  %ebp
    ret
.Ltmp0:
    .size   main, .Ltmp0-main

    .type   .L.str,@object        # @.str
    .section .rodata.str1.1,"aMS",@progbits,1
.L.str:
    .asciz  "hello, world!\n"
    .size   .L.str, 15

    .section ".note.GNU-stack","",@progbits

```

10000011	00000001	00010001	00000000	00111101	11111100	01110100	00111101
00000000	01000000	00000000	00000000	00000000	00000000	00000000	00000000
10010000	00000000	00000000	00000000	01010000	00000000	00000111	00110000
00001011	00000001	00001011	00000011	00001010	00000000	00000000	00000000
00000000	00100000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00100000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
01110000	00010000	00000000	00100000	00000001	00000000	00000000	00000000
00000000	00000000	00000000	00100000	00000001	00000000	00000000	00000000
00000000	00000000	00000000	01000000	00000001	00000000	00000000	00000000
00000000	00100000	00000000	01000000	00000001	00000000	00000000	00000000
11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
10010000	10000000	00000000	01000000	00000001	00000000	00000000	00000000
00101110	01100100	01111001	01101110	01100001	01101101	01101001	01100011
10110000	00000100	00000000	00100000	00000001	00000000	00000000	00000000
10110000	00000100	00000000	00100000	00000001	00000000	00000000	00000000
10100000	00000001	00000000	00000000	00000000	00000000	00000000	00000000
10110000	00000100	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00100000	00000000	00000000

...

hello.c
[uses printf]



compile

assembly code
for hello.c



assemble

```
01101001 01001010 10000010 11011001  
00101011 01100000 10001011 10111010
```

hello.c
[uses printf]

stdio.h
[describes printf]



stdio.c
[defines printf]



compile

assembly code
for hello.c



compile

assembly code
for stdio.c



assemble

```
01101001 01001010 10000010 11011001  
00101011 01100000 10001011 10111010
```



assemble

```
01101001 01001010 10000010 11011001  
00101011 01100000 10001011 10111010
```

link



```
01101001 01001010 10000010 11011001  
00101011 01100000 10001011 10111010
```

CS50 Lunch

cs50.net/rsvp

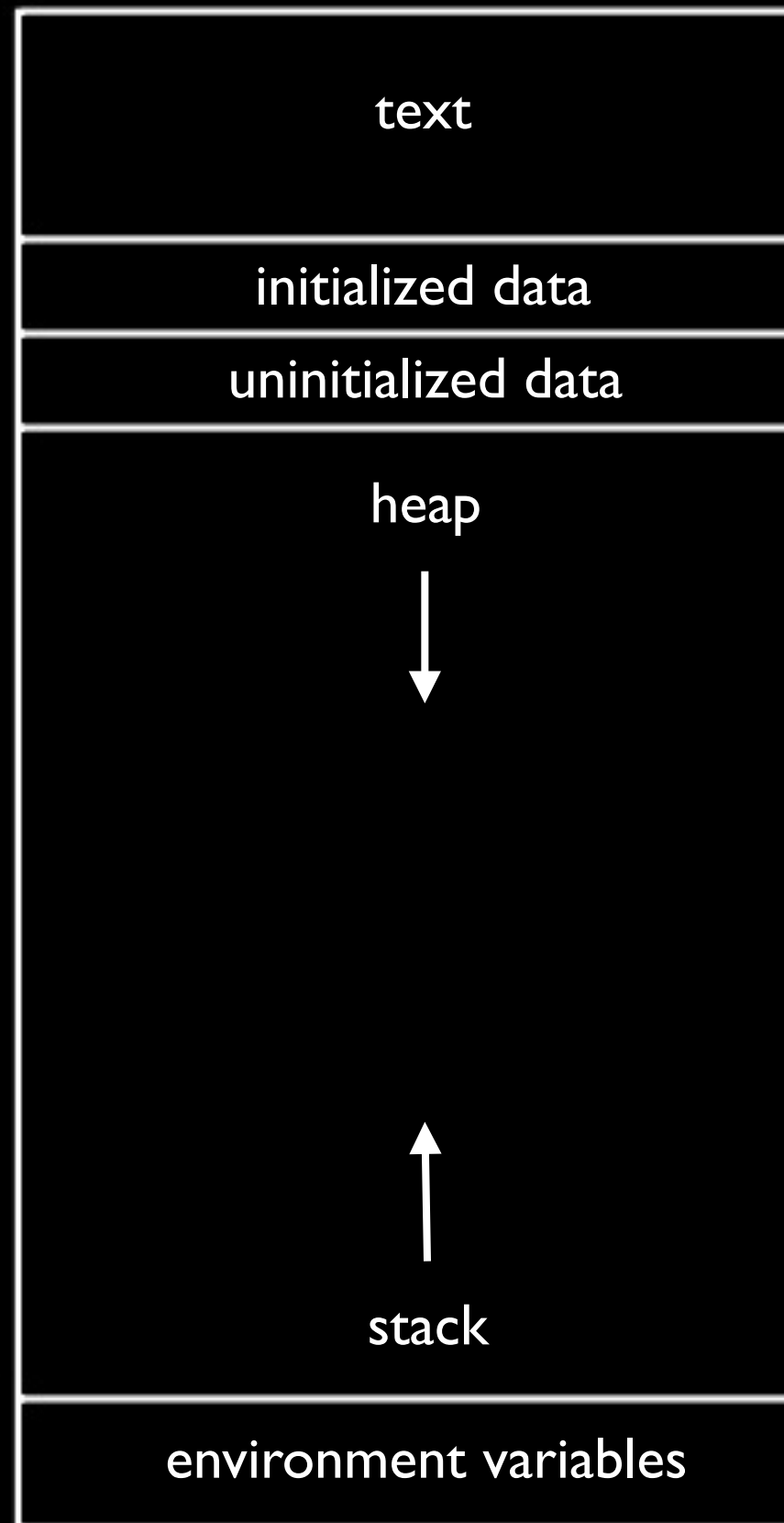
no lecture Mon 10/14

quiz 0 on Wed 10/16

review session on Mon 10/14

lecture on Fri 10/18

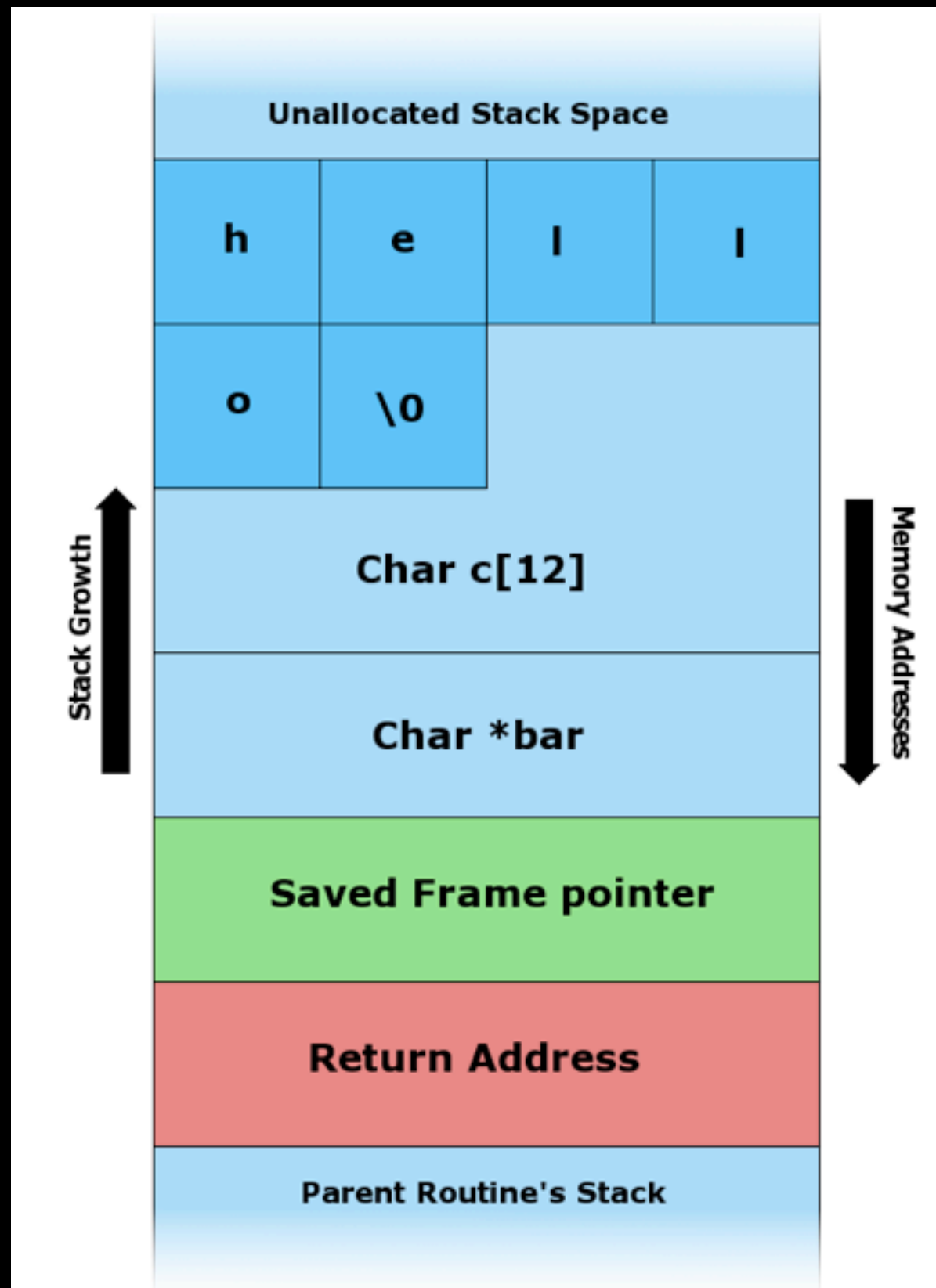
no lecture on Fri 10/18



```
#include <string.h>

void foo(char* bar)
{
    char c[12];
    memcpy(c, bar, strlen(bar));
}

int main(int argc, char* argv[])
{
    foo(argv[1]);
}
```



malloc

valgrind

```
valgrind --leak-check=full ./program
```

```
Invalid write of size 4
```

```
at 0x804840F: f (memory.c:21)
```

```
by 0x8048421: main (memory.c:26)
```

```
40 bytes in 1 blocks are definitely lost in loss record 1 of 1
```

```
at 0x4025BDC: malloc (vg_replace_malloc.c:195)
```

```
by 0x8048405: f (memory.c:20)
```

```
by 0x8048421: main (memory.c:26)
```

free

CS50 Library

GetChar

GetDouble

GetFloat

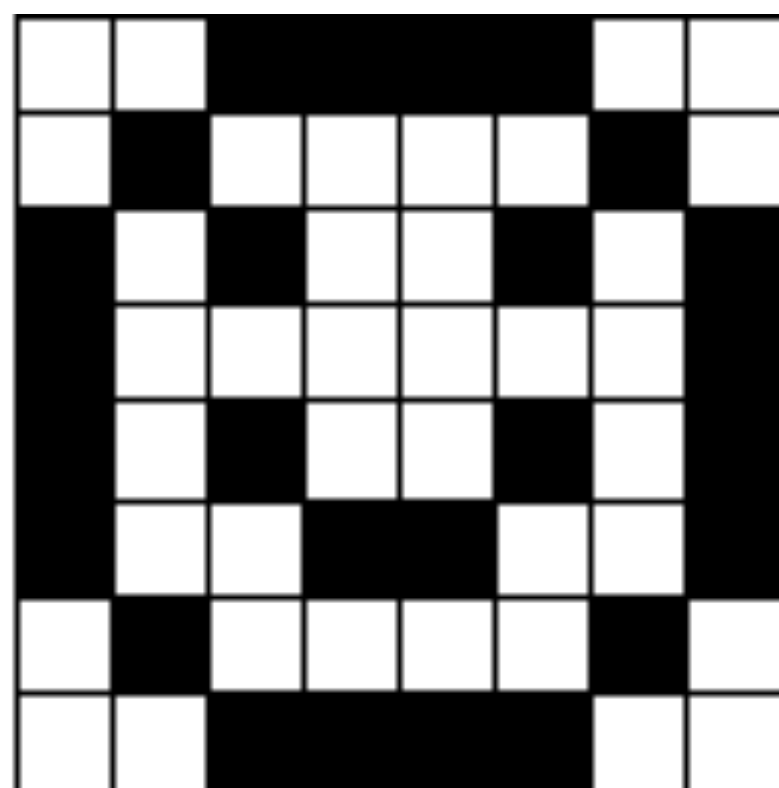
GetInt

GetLongLong

GetString

```
typedef struct
{
    int id;
    string name;
    string house;
}
student;
```

```
11000011
10111101
01011010
01111110
01011010
01100110
10111101
11000011
```



offset	type	name	
0	WORD	bfType	} BITMAPFILEHEADER
2	DWORD	bfSize	
6	WORD	bfReserved1	
8	WORD	bfReserved2	
10	DWORD	bfOffBits	
14	DWORD	biSize	} BITMAPINFOHEADER
18	LONG	biWidth	
22	LONG	biHeight	
26	WORD	biPlanes	
28	WORD	biBitCount	
30	DWORD	biCompression	
34	DWORD	biSizeImage	
38	LONG	biXPelsPerMeter	
42	LONG	biYPelsPerMeter	
46	DWORD	biClrUsed	
50	DWORD	biClrImportant	} RGBTRIPLE
54	BYTE	rgbtBlue	
55	BYTE	rgbtGreen	
56	BYTE	rgbtRed	} RGBTRIPLE
57	BYTE	rgbtBlue	
58	BYTE	rgbtGreen	
59	BYTE	rgbtRed	} RGBTRIPLE
...			
243	BYTE	rgbtBlue	
244	BYTE	rgbtGreen	
245	BYTE	rgbtRed	



to be continued...