Basic Program Structure (Asm + C)

Standard form Asm

label: opcode operands comment

labels name current position in code
current pos is addr at which next inst goes

for: add $1, $eax
     sib $1, $ebx
     bec for jbr: t foo if zero not set

Directives are aimed at the assembler
equiv MAX_CNT 12
mov $MAX_CNT, $ebx
movl 12, $ebx
pseudo-op
MAX_CNT .eq 12

.text

tells assembler we are in text (code) section
other common section names include

.ro data read only data
.data initialized data
.bss uninitialized data

comes from IBM 704 assembly day, late 1950s
"black storage start"

distinct sections useful in embedded systems

alternative syntax
section name
labels important in data section

str: .string "this is an example"
var: .int 3
it is label that means "right here"

&x80 &+& 4  means 4 addressing beyond current position

add — x x x x

org — (

sub — x x x x x x

C program structure

classically, c goes into two sets of files

— c program files

— h header files

/*

* @

* global comments

# define MAX_CNT 12 // preprocessor dir
// bar.c

// header comment

#include <stdio.h>  // <> system
#include "bar.h"  // <> local

int var = 1;  // global var

int *x;

main()
{
    int var = 2;  // local var

    // code
}

?