A Simple Assembly Language

August 25

CSC201 Section 002

Fall, 2000

Directives and Executable Instructions

- "Directives" or "pseudo-ops": control what the Assembler does
  - Not used after the assembly process
- Executable instructions, or "code": assembled into the output (executable program)

A Program Template

```
; This is a template you can use to write your SASM programs

.486                        ; use the Intel 486 instruction set
.model  flat, stdcall       ; "flat" memory model, standard parameter
passing to subroutines

.stack  1000h               ; initialize stack size to 4096 bytes (16^3)

include sasmacros.inc       ; this file defines the SASM instructions
title  template  program ; for documentation purposes only

.data                       ; start the data declaration part of the program
---data definitions go here---

.code                  ; start the executable instructions of the program
main:                  ; this label defines where program execution will start
---code goes here---

end                         ; signals the assembler this is the end of the file
```

Labels

- Alphabetical and numerical characters, starting with a letter
- Case insensitive

The .DATA Section

- Defines...
  - how much memory to allocate for data used by the program
  - where in memory each data item is located
  - how that memory is initialized before the program starts execution

Defining Characters and Strings

- DB = "Define one Byte of memory"

```
char0    db  ?
char1    db  20
char2    db  14h
char3    db  00010100b
char4    db  'x'
string1  db  "abc"
string2  db  20, 14h, 00010100b, 'x', "abc"
string3  db  "Hello, world!", 0ah, 0
```

Defining Numerical Values

- DD = "Define one Doubleword (4 bytes) of memory"

```
doubleword0  dd  ?
doubleword1  dd  25
doubleword2  dd  -2037
doubleword3  dd  1.234
doubleword4  dd  -3.6247e2
```

Duplicating or Repeating Values

```
string4     db  5 dup 'a'
doubleword5  dd  1000 dup ?
strarray    db  4 dup ("hi, mom!", 0ah, 0)
```

A "Hello, World!" Program

```
; This is a template you can use to write your SASM programs

.486                        ; use the Intel 486 instruction set
.model  flat, stdcall       ; "flat" memory model, standard parameter
passing to subroutines

.stack  1000h               ; initialize stack size to 4096 bytes (16^3)

include sasmacros.inc     ; this file defines the SASM instructions
title  template  program ; for documentation purposes only

.data                       ; start the data declaration part of the program
string1    db   "Hello, world!", 0ah, 0

.code                ; start the executable instructions of the program
main:                ; this label defines where program execution will start
        put_str  string1        ; macro for outputting a zero-terminated
string
        done                 ; do some "housecleaning" and halt execution



end                         ; signals the assembler this is the end of the file
```