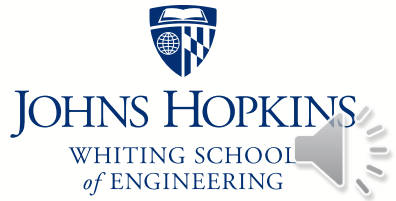
Johns Hopkins Engineering

Module 5: MIPS Assembly Programming

EN605.204: Computer Organization



MIPS Assembly Program Structure

- The format of a MIPS assembly program consists of two basic parts
  - data section (where data goes)
  - text section (where code goes)

MIPS Assembly Program Structure

- Within the data and text sections there are additional elements important to the assembly program

  - **Comments**
    - "#" character – anything after the "#" is consider a comment
  - **Directives**
    - begin with "."
    - non-executable messages to the assembler that aid in completing the assembly process
    - Example : .data or .text respectively start and end data declarations and procedures / functions

MIPS Assembly Program Structure

- Data Declarations
  - The .data section of the program is where data must be declared
  - Similar to defining / initializing variables in a high-level language
  - General format

<variable name>: .<data type> <initial value>

MIPS Assembler Directives

Declaration	
`.byte`	8-bit variable(s)
`.half`	16-bit variable(s)
`.word`	32-bit variable(s)
`.ascii`	ASCII string
`.asciiz`	NULL terminated ASCII string
`.float`	32 bit IEEE floating-point number
`.double`	64 bit IEEE floating-point number
`.space <n>`	<n> bytes of uninitialized memory

MIPS Assembler Directives

- Integer declarations

32 bit

```
variable1:      .word       220000
```

16 bit

```
variable1:      .half       2000
```

8 bit

```
variable1:      .byte       2
```

MIPS Assembler Directives

- Characters / String declarations

```
stringvar1:     .asciiz     “This is a sample null terminated string\n”
stringvar2:     .ascii      “This is a sample string without a null termination”
```

- Floating-Point declarations

32 bit

```
pi:             .float      3.1415
```

64 bit

```
bigpi:          .double     3.14159265359….
```

MIPS Assembly