TEXTEQU Directive

1. TEXTEQU Directive Overview:
The TEXTEQU directive in assembly language is used to create text macros. These text macros are like placeholders for text or expressions that you can reuse in your code. There are three common formats for the TEXTEQU directive:

• **name TEXTEQU <text>**: This format assigns name as a text macro and associates it with the specified text enclosed in angle brackets < >.
• **name TEXTEQU textmacro**: Here, name becomes a text macro, and it takes on the content of an existing text macro called textmacro.
• **name TEXTEQU %constExp**r: This format assigns name as a text macro and associates it with a constant integer expression %constExpr.

```
name TEXTEQU <text>
name TEXTEQU textmacro
name TEXTEQU %constExp
```

Let's consider an example where we create a text macro for a prompt message:

```
continueMsg TEXTEQU <"Do you wish to continue (Y/N)?"">
```

Here, continueMsg is associated with the text "Do you wish to continue (Y/N)?". This text macro can

be used to simplify the creation of prompt messages in your code.

Building Text Macros: Text macros can also be used to build more complex macros by combining them. For example:

```
rowSize = 5
count TEXTEQU %(rowSize * 2)
move TEXTEQU <mov>
setupAL TEXTEQU <move al,count>
```

Here, count is calculated based on the value of rowSize, and move is associated with the text "mov". Then, setupAL is built using move and count. So, setupAL becomes "mov al,10" because count is calculated as rowSize * 2.

Redefining Text Macros: Unlike the EQU directive, a symbol defined with TEXTEQU can be redefined at any time within the same source code file. This means you can change the text or expression associated with a text macro as needed.

In summary, the TEXTEQU directive is used to create text macros in assembly language, which are like placeholders for text or expressions that you can reuse. These macros can simplify code and make it more readable. Unlike symbols defined with EQU, text macros can be redefined in the same source file, allowing for flexibility in your code.