

Session – 26 : Repeat session on make

23-10-2019, 11:00 – 12:00 Hrs, RJN 302

Here are the steps followed and the interpretation.

Situation	Interpretation
Directory does not have any makefile	The make command will crib that no targets specified and no makefile is found. It is looking for a file called “Makefile” in the directory.
Directory contains an empty file called Makefile.	The make command will crib that there are no targets defined. The file “Makefile” should contain a default target.
The “Makefile” contains a single line as follows: <code>default:</code>	The make command will say that there is nothing to be done for default target.
The “Makefile” contains the following two lines: <code>default:</code> <code> echo “Hello”</code>	The make command will print out the following two lines: <code>echo “Hello”</code> <code>Hello</code> This means the “rule” mentioned below the target is first printed out to the screen and then executed.
The “Makefile” contains the following two lines: <code>default:</code> <code> @echo “Hello”</code>	The make command will print out the following line: <code>Hello</code> This means the “rule” mentioned below the target is not printed out to the screen but simply executed. The @ character at the beginning of the “rule” conveys that the rule itself need not be printed out to the screen.
The “Makefile” contains the following two lines: <code>default: xaa</code> <code> @echo “Hello”</code>	The make command will crib that there is no rule specified to make target “xaa”, needed by the target “default”. The keyword xaa could imply a target or a file in the same directory.

<p>The directory contains an empty file called <code>xaa</code> and the Makefile contains the following two lines:</p> <pre>default: xaa @echo "Hello"</pre>	<p>The make command will print out the following line:</p> <p><code>Hello</code></p> <p>The dependency <code>xaa</code> is satisfied by the presence of the file <code>xaa</code> in the directory.</p>
<p>The directory does not contain a file called <code>xaa</code> but the Makefile contains the following lines:</p> <pre>default: xaa @echo "Hello" xaa: @echo "Hi"</pre>	<p>The make command will print out the following lines:</p> <p><code>Hi</code> <code>Hello</code></p> <p>The dependency <code>xaa</code> is <i>first</i> made by the rule given below it. This rule printed out the word <code>Hi</code>. Then the default target is made by executing the rule given below it. This rule printed out <code>Hello</code>.</p>
<p>The directory does not contain a file called <code>xab</code> but the Makefile contains the following lines:</p> <pre>default: xaa xab @echo "Hello" xaa: @echo "Hi"</pre>	<p>The make command will crib that there is no rule to make the target <code>xab</code> as needed by the target "<code>default</code>". It does not execute any other rules.</p> <p>This means each of the dependencies is checked for either the files of those names in the directory or targets with those names in the Makefile. If any dependency is missed out, make utility does not proceed further.</p>

Other features of the Makefile syntax are embellishments to make alises to expand to the list of dependencies, variable names to use in a ifeq-endif loop etc.,

The make utility is originally used to compile programs such that the compiler commands and options are customized for different locations where the make utility is invoked. However, the framework of Makefile could be used for any action.

For example, you can use the Makefile to compile LaTeX documents to make dvi or ps or pdf files out of your tex documents. You can use the Makefile to make tarballs or zip files of your important folders and copy them over to safe backup locations.

The recursive nature of the target that could be one of the dependencies helps in creating a heirarchical set of actions / rules to perform a complex task.