Input and output Redirection

Input redirection on a UNIX/Linux system allows a program to read the expected input from an input file instead of from the keyboard (standard input stream - cin).

To create a file for use with input redirection, a text file is edited and the input that would normally be typed in when a program is run is put into the input file. For a program that requires three lines of data to be entered, create an input file containing the three lines and then use it with input redirection.

Example. Assume the executable Project_05 (executable of Project_05.cpp) requires the user to enter in three sentences. The three sentences are instead placed in the file P5_in1.txt. Now the executable Project_05 can be run as shown below:

blackhawk> **./Project_05 < P5_in1.txt**

The user does not have to enter in any information. All values read using cin (the standard input stream) are now read from the input file instead of the keyboard.

Output redirection on a UNIX/Linux system allows a program to write output to a file instead of to the terminal (standard output stream – cout).

With output redirection all terminal output is written to the user specified output file. As long as the user has write permissions for the directory where the file is saved, the file is created.

Example: The executable Project_05 (executable of Project_05.cpp) writes information to the terminal window (uses cout). To send this information to a file instead, use output redirection as shown below:

blackhawk> **./Project_05 > P5_out1.txt**

With output file redirection in place, the user supplies any input values required using the standard input stream (cin) device, which is the keyboard. To complicate this procedure, the user needs to know what the prompts are and what information has to be entered because all prompts are redirected to the output file instead of the terminal.

Input/output redirection combined results in all input being read from an input file and all output going to an output file; thereby solving the problem of users entering information with output redirection.

Example: The executable Project_05 (executable of Project_05.cpp) requires user input from the keyboard and writes information to the terminal window. To read the information from the input file P5_in1.txt and send the output information to the output file P5_out1.txt, use both input and output redirection at the same time.

blackhawk> **./Project_05 < P5_in1.txt > P5_out1.txt**
blackhawk>

The file P5_out1.txt contains all information that normally would be written to the terminal.