

CSC 357 Program 3 Test Plan

This is an input/output test plan for programming assignment 3. In this plan, the only input file used is `Smakefile`. To allow different versions of `Smakefile`, the test plan creates a symbolic link to a particular version of `Smakefile`. These linked-to files are listed in the **In File** column of the plan. The test plan also lists the command-line arguments, if any.

The output is in three forms -- `stdout`, `stderr`, and files generated by `Smakefile` actions. These are listed in the three columns of the test plan.

In addition to the different versions of the `Smakefile`, there are a number of other input files that must be present. These include the `Makefile` and the `Smakefile` rules. You must copy the entire contents of this testing directory, and its subdirectories, in order for the test plan to be properly executed.

The test-execution script is `run.csh`. It assumes a program named `smake` exists in this directory, as well as all necessary input files and subdirectory. The script must be run on `hornet` in order to produce behavior consistent with the files in the expected-output directory.

NOTE: Unlike previous run scripts, this one echos program output as it progresses. In addition, the output files are created and differ from the expected output files.

Case	In File	argv	stdout	stderr
1	no-actions			
2	mixed-spacing			
3	1-rule		<pre>r1 action echo "r1 action"</pre>	
4	2-rules		<pre>touch f1.txt 0 f1.txt</pre>	
5	2-rules		<pre>0 f1.txt</pre>	
6	self		<pre>gcc -Wall -g -c smake.c gcc -Wall -g -c util.c gcc -Wall -g -c list.c gcc -Wall -g -c hash.c gcc -Wall -g -c parse.c gcc -Wall -g -o smake_t ...</pre>	
7	self		<pre>smake: Nothing to be done for 'all'.</pre>	

CSC357-W07-Prog3-test-plan

8	self	clean	rm -f *.o smake_t	ls: No match.
9	self	print	In make-print. csh -c ./make-print.csh	ls: No match.
10	self	hash.o	gcc -Wall -g -c hash.c 2 hash.o	
11	self	all	gcc -Wall -g -c smake.c gcc -Wall -g -c util.c gcc -Wall -g -c list.c gcc -Wall -g -c parse.c gcc -Wall -g -o smake_t ... 2 hash.o 2 list.o 2 parse.o 4 smake.o 2 util.o	
12	multi-actions		touch f3.txt rm -f f1?.txt touch f2.txt touch f15.txt touch f1.txt touch f10.txt touch f11.txt touch f12.txt touch f13.txt touch f14.txt 0 f1.txt 0 f10.txt 0 f11.txt 0 f12.txt 0 f13.txt 0 f14.txt 0 f15.txt 0 f2.txt 0 f3.txt 0 f4.txt	

13	multi-actions	<pre>CASE 13: 0 f1.txt 0 f10.txt 0 f11.txt 0 f12.txt 0 f13.txt 0 f14.txt 0 f15.txt 0 f2.txt 0 f3.txt 0 f4.txt</pre>
14	writeup	<pre>CASE 14: main built other built Echo Done! gcc -c main.c echo "main built" gcc -c other.c echo "other built" echo "Echo" gcc -o main main.o other.o echo "Done!" 2 main.o 2 other.o</pre>
15	writeup	<pre>main built other built Echo Done! gcc -c main.c echo "main built" gcc -c other.c echo "other built" echo "Echo" gcc -o main main.o other.o echo "Done!"</pre>

CSC357-W07-Prog3-test-plan

16	4-deep	clean	/bin/rm -f r? r??	
17	4-deep		r5 unconditional action r6 unconditional action /bin/touch r8.txt /bin/touch r8 /bin/touch r9.txt /bin/touch r9 echo "r5 unconditional action" echo "r6 unconditional action" /bin/touch r3 /bin/touch r1	
18	4-deep		r5 unconditional action r6 unconditional action echo "r5 unconditional action" echo "r6 unconditional action" /bin/touch r3 /bin/touch r1	
19	empty			smake: no rules ... exiting!
20	empty-name			smake: 1: empty name
21	malformed-rule			smake: malformed rule on line
22	no-tab			smake: malformed rule on line
23	missing-rule			smake: *** No rule to make target 'a', needed by 'x'. Stop.
24	not-found			sh: nonexistent: command not found smake: [x] Error 1 nonexistent
25	not-executable			sh: ./nonexecutable: Permission denied smake: [x] Error 1 nonexecutable

CSC357-W07-Prog3-test-plan

26	unexpected-action	
27		1 2 3

smake: unexpected action on line 1
usage: smake [rule]