CS61B, Fall 2001 **Lab/HW #11** **P. N. Hilfinger**

Due: Fri., 7 December 2001 at midnight

Create a directory to hold your answers to this homework set. Copy the files from `$master/hw/hw11` into this directory. Use the command `submit hw11` to submit your solutions to the problems below.

1. Write a Java program that duplicates part of the functionality of `gmake`. The input (a "makefile") contains blank lines (which are ignored), pure dependency lines like this:

$NAME_0$: $NAME_1 \cdots NAME_n$

(where $n \geq 0$) which means "$NAME_0$ (the *target*) depends on $NAME_1 \cdots NAME_n$," and pairs of lines like this:

$NAME_0$: $NAME_1 \cdots NAME_n$
COMMAND

which has the same meaning as a pure dependency line, plus "to build $NAME_0$, first build everything it depends on, and then issue *COMMAND*." There may be at most one command for each distinct target. There may be any number of additional pure dependency lines. The order in which dependency lines or dependency line/command line pairs appear is irrelevant. You can essentially ignore any *NAME* that never appears as a target.

When you start your program with the command

`java make -f` *FILE NAME*

it is supposed to read in the makefile named *FILE,* and then print a proper sequence of *COMMANDs* to build *NAME* (there may be more than one valid sequence).

See the template file `~cs61b/hw11/make.java`. To make the next problem easier, I suggest that you use the `ucb.io` package to handle the input and output.

2. Translate the program you wrote for the preceding exercise into C. See the template file `~cs61b/hw11/make.c`. To debug your C program, I suggest using the `M-x gdb` command in Emacs, which should give you a rather familiar debugging interface.