2.3 How make Processes a Makefile

By default, make starts with the first target (not targets whose names start with '.'). This is called the default goal. (Goals are the targets that make strives ultimately to update. You can override this behavior using the command line (see Arguments to Specify the Goals) or with the .DEFAULT_GOAL special variable (see Other Special Variables).

默认情况下，`make` 从第一个目标开始（不是名称以 “.” 开头的目标）。 这称为默认目标。 （目标是 make 最终努力更新的目标。您可以使用命令行（请参阅指定目标的参数）或使用 .DEFAULT_GOAL 特殊变量（请参阅其他特殊变量）覆盖此行为。

In the simple example of the previous section, the default goal is to update the executable program edit; therefore, we put that rule first.

在上一节的简单示例中，默认目标是更新可执行程序 edit； 因此，我们将这条规则放在首位。

Thus, when you give the command:

```
make
```

make reads the makefile in the current directory and begins by processing the first rule. In the example, this rule is for relinking edit; but before make can fully process this rule, it must process the rules for the files that edit depends on, which in this case are the object files. Each of these files is processed according to its own rule. These rules say to update each '.o' file by compiling its source file. The recompilation must be done if the source file, or any of the header files named as prerequisites, is more recent than the object file, or if the object file does not exist.

因此，当你给出命令：

```
make
```

`make` 读取当前目录中的 `makefile` 并从处理第一条规则开始。 在示例中，此规则用于重新链接 edit； 但是在 make 可以完全处理这个规则之前，它必须处理 edit 所依赖的文件的规则，在这种情况下是目标文件。 这些文件中的每一个都根据自己的规则进行处理。 这些规则说通过编译其源文件来更新每个 “.o” 文件。 如果源文件或任何作为先决条件命名的头文件比目标文件更新，或者目标文件不存在，则必须进行重新编译。

The other rules are processed because their targets appear as prerequisites of the goal. If some other rule is not depended on by the goal (or anything it depends on, etc.), that rule is not processed, unless you tell make to do so (with a command such as make clean).

处理其他规则是因为它们的目标显示为目标的先决条件。 如果目标不依赖其他规则（或它所依赖的任何东西等），则不会处理该规则，除非您告诉 make 这样做（使用诸如 make clean 之类的命令）。

Before recompiling an object file, make considers updating its prerequisites, the source file and header files. This makefile does not specify anything to be done for them—the '.c' and '.h' files are not the targets of any rules—so make does nothing for these files. But make would update automatically generated C programs, such as those made by Bison or Yacc, by their own rules at this time.

在重新编译目标文件之前，make 会考虑更新其先决条件、源文件和头文件。 这个 makefile 没有指定要为它们做的任何事情 —— ".c" 和 ".h" 文件不是任何规则的目标——所以 make 对这些文件什么都不做。 但是此时 make 会按照自己的规则更新自动生成的 C 程序，例如 Bison 或 Yacc 制作的 C 程序。

After recompiling whichever object files need it, make decides whether to relink edit. This must be done if the file edit does not exist, or if any of the object files are newer than it. If an object file was just recompiled, it is now newer than edit, so edit is relinked.

在重新编译任何需要的目标文件后，make 决定是否重新链接 edit 。 如果文件 edit 不存在，或者任何目标文件比它新，则必须这样做。 如果一个目标文件刚刚被重新编译，它现在比 edit 新，所以编辑被重新链接。

Thus, if we change the file insert.c and run make, make will compile that file to update insert.o, and then link edit. If we change the file command.h and run make, make will recompile the object files kbd.o, command.o and files.o and then link the file edit.

因此，如果我们更改文件 insert.c 并运行 make，make 将编译该文件以更新 insert.o，然后进行链接 edit 。 如果我们更改文件 command.h 并运行 make，make 将重新编译目标文件 kbd.o、command.o 和 files.o，然后链接文件 edit 。
