

# Programming Tricks, Part 1

Jacob K LaGrou edited this page on Feb 5 · 9 revisions

## Use `cat` as your IDE

Who needs an editor? IDE? We can just use `cat` ! You've seen `cat` being used to read the contents of files but it can also be used to read the standard-input and send it back to standard output.

```
$ cat
HELLO
HELLO
```

To finish reading from the input stream close the input stream by pressing `CTRL-D`

Let's use `cat` to send standard input to a file. We will use `>` to redirect its output to a file:

```
$ cat > myprog.c
#include <stdio.h>
int main() {printf("Hi!");return 0;}
```

(Be careful! Deletes and undos are not allowed...) Press `CTRL-D` when finished.

## Edit your code with `perl` regular expressions (aka "remember your perl pie")

A useful trick if you have several text files (e.g. source code) to change is to use regular expressions. `perl` makes this very easy to edit files in place. Just remember 'perl pie' and search on the web...

An example. Suppose we want to change the sequence "Hi" to "Bye" in all `.c` files in the current directory. Then we can write a simple substitution pattern that will be executed on each line at time in all files:

```
$ perl -p -i -e 's/Hi/Bye/' *.c
```

(Don't panic if you get it wrong, original files are still there; they just have the extension `.bak`) Obviously there's a lot more you can do with regular expressions than changing Hi to Bye.

## Use your shell `!!`

Edit

New Page

▼ Pages 51

[Home](#)

[#Example Markdown](#)

[#Informal Glossary](#)

[#Piazza: When And How to Ask For Help](#)

[C Programming, Part 1: Introduction](#)

[C Programming, Part 2: Text Input And Output](#)

[C Programming, Part 3: Common Gotchas](#)

[C Programming, Part 4: Debugging](#)

[Deadlock, Part 1: Resource Allocation Graph](#)

[Deadlock, Part 2: Deadlock Conditions](#)

[File System, Part 1: Introduction](#)

[File System, Part 2: Files are inodes \(everything else is just data...\)](#)


[File System, Part 3: Permissions](#)

[File System, Part 4: Working with directories](#)

[File System, Part 5: Virtual file systems](#)

[Show 36 more pages...](#)

Clone this wiki locally

 Clone in Desktop

To re-run the last command just type `!!` and press `return` To re-run the last command that started with g type `!g` and press `return`

## Use your shell `&&`

Tired of running `make` or `gcc` and then running the program if it compiled OK? Instead, use `&&` to chain these commands together

```
$ gcc program.c && ./a.out
```

## Make can do more than make

You might also try putting a line in your Makefile that will compile, and then run your program.

```
run : $(program)
    ./${$(program)}
```

Then running

```
$ make run
```

will make sure any changes you've made are compiled, and run your program in one go. Also good for testing many inputs at once. Although you probably would just rather write a regular shell script for that.

## Is your neighbor too productive? C pre-processors to the rescue!

Use the C pre-processor to redefine common keywords e.g.

```
#define if while
```

Protip: Put this line inside one of the standard includes e.g. `/usr/include/stdio.h`

## Who needs functions when you C have the preprocessor

OK, so this is more of a gotcha. Be careful when using macros that look like functions...

```
#define min(a,b) a<b?a:b
```

A perfectly reasonable definition of a minimum of a and b. However the pre-processor is just a simple text wrangler so precedence can bite you:

```
int value = -min(2,3); // Should be -2?
```

Is expanded to

```
int value = -2<3 ? 2 :3; // Ooops.. result will be 2
```

A partial fix is to wrap every argument with `()` and also the whole expression with `()`:

```
#define min(a,b) ( (a) < (b) ?(a):(b) )
```

However this is still *not* a function! For example can you see why `min(i++,10)` might increment i once or twice!?

Legal and Licensing information: Unless otherwise specified, submitted content to the wiki must be original work (including text, java code, and media) and you provide this material under a [Creative Commons License](#). If you are not the copyright holder, please give proper attribution and credit to existing content and ensure that you have license to include the materials.

