Tip 6: Aggregate your includes

Ben Klemens

11 October 2011

level: casual user
purpose: stop thinking about standard header files

There was once a time when compilers took several seconds or minutes to compile even relatively simple programs, so there was human-noticeable benefit to reducing the work the compiler has to do. My current copies of `stdio.h` and `stdlib.h` are about 1,000 lines long [try `wc -l /usr/include/stdlib.h`] and `time.h` another 400, meaning that

```c
#include <time.h>
#include <stdio.h>
#include <stdlib.h>
int main(){
    srandom(time(NULL));
    printf("%li\n", random());
}
```

is actually a ∼2,400-line program.

You see where this is going: your compiler doesn't think 2,400 lines is a big deal anymore, and this compiles in under a second. So why are we spending time picking out just the right headers for a given program?

To do:
Write yourself a single header, let us call it `allheads.h`, and throw in every header you've ever used, so it'll look something like:

```c
#include <math.h>
#include <time.h>
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <gsl/gsl_rng.h>
```

I can't tell you exactly what it'll look like, because I don't know exactly what you use day to day.

Now that you have this aggregate header, you can just throw one

```c
#include <allheads.h>
```

on top of every file you write, and you're done with thinking about headers. Sure, it will expand to perhaps ten thousand lines of extra code, much of it not relevant to the program at hand. But you won't notice.

Having junk in the name space may have conesequences from time to time. Say that you've defined a function called `rndm`, and at some point you forget the compressed name and call `random`. As above, `random` is a real function declared in `stlib.h`, so you won't get the usual warning about using an undeclared function. I don't count this as all that much of a problem.

When I wrote the Apophenia library, by the way, I had in mind this kind of thing, so I have

```
#include <apop.h>
```

at the head of every program I write, whether I'll be using anything from the Apophenia library or not, because that includes `stdio`, `stdlib`, the stats-relevant parts of the GSL, et cetera. And then I more-or-less never think about headers.

The next tip will show how to eliminate even that `#include <allheads.h>` line at the top of all your programs.