

Tip 80: Send in parameters fast

Ben Klemens

10 March 2012

level: your program has options for users

purpose: Get options with zero overhead

OK, there isn't *zero* overhead: you will need

```
#include <stdlib.h>
```

at the top of your program. Once you have that, you can get environment variables from the user with `getenv`.

For the purposes of an example, let us print a message to the screen as often as the user desires. The message is set via the environment variable `msg` and the number of repetitions via `reps`. Notice how we set defaults for both (at 10 and "Hello.") should `getenv` return `NULL`.

```
#include <stdlib.h> //getenv
#include <stdio.h>  //printf

int main() {
    char *repstext=getenv("reps");
    int reps = repstext ? atoi(repstext) : 10;

    char *msg = getenv("msg");
    if (!msg) msg = "Hello.";

    for (int i=0; i< reps; i++)
        printf("%s\n", msg);
}
```

Usage, if the program compiled to `a.out`:

```
reps=10 msg="Ha" ./a.out
msg="Ha" ./a.out
reps=20 msg=" " ./a.out
```

The trick here on the command line is that you can set environment variables to be sent to a program on the same command line as the program call. You might find this

to be odd—the inputs to a program should come *after* the program name, darn it—but the oddness aside, you can see that it took roughly zero set up within the program itself, and we get to have named parameters on the command line almost for free.

This is a nice way to get parameters in quickly, and then when your program is a little further along, you can use `getopt` to do them the usual way.