Re: FIbonacci

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2005-12/msg01670.html

- *From*: mwojcik@xxxxxxxxxxx (Michael Wojcik)
- *Date*: 13 Dec 2005 18:40:39 GMT

In article <z7adnct6kL7RAwHeRVn-ig@xxxxxxxxxxx>, Eric Sosman <esosman@xxxxxxxxxxxxxxxxxxx> writes:
> Peteris Krumins wrote:
> >
> > #include <stdio.h>
> >
> > unsigned fib(unsigned n) {
> > return n <= 2 ? 1 : fib(n - 1) + fib(n - 2);
> > }
> >
> > int main(void) { printf("fib(17) is: %d\n", fib(17)); return 0; }
>
> Somebody always proposes this solution, but it's a
> poor one. Try it with fib(47), say, and tell us how
> long it takes. Hint: You won't need a high-precision
> timer.

Well, there's always this one (with error checking left as an
exercise for the reader):

```
/***
prev2 and prev are the two immediately preceeding values in the
series; prev2 is F(n-2) and prev is F(n).
pos is the current position in the series.
want is the position we're looking for.
***/

unsigned fib_cps(unsigned prev2, unsigned prev, unsigned pos, unsigned want)
{
if (want <= 2) return 1;
if (pos == want) return prev2 + prev;
return fib_r(prev, prev2 + prev, pos + 1, want);
}

unsigned fib(n)
{
return fib_cps(1, 1, 3, n);
}
```

Even written this way, and compiled without optimization (I verified that the compiler left the tail−recursive call in rather than optimizing it to a branch), this computes fib(47) in negligible time. But of course it's just the forward−iterative version written as tail recursion.

It'd be possible to rewrite Peteris' backward−recursing algorithm using continuation−passing style and tail recursion, but since C lacks dynamic closures we'd have to emulate them. And that would just involve recursively creating some sort of list of instructions to run the forward−iterative algorithm, so it's not very interesting.

−−
Michael Wojcik michael.wojcik@xxxxxxxxxxxxxx

Is it any wonder the world's gone insane, with information come to be the only real medium of exchange? −− Thomas Pynchon
.

- ***Follow−Ups***:
  - ♦ ***Re: FIbonacci***
    - ◊ *From:* Tim Rentsch

- ***References***:
  - ♦ ***FIbonacci***
    - ◊ *From:* MARQUI