Example: Finding Prime Numbers

Ned Nedialkov

McMaster University
Canada

SE 3F03
January 2013

Outline

Finding primes

Find all prime numbers up to a given N

- For each odd number $\leq N$, find its factors (2 is prime),
- If no factor can be found, then it is prime
- If there is a factor, it must be odd
- How to find factors?

Trial division

- Given an odd number M, check if it can be factored
- Check if 3, 5, 7, ... divide M
- Check the odd numbers up to $\sqrt{M}$
  - if M is divisible by some m then $M = m \times q$
  - if $q < m$, M would have been detected earlier as divisible by q
- Examples
  - $27 \mod 3 = 0$, not prime
  - $89 \mod 3 = 2$, $89 \mod 5 = 4$, $89 \mod 7 = 5$, $89 \mod 9 = 8$ do not check 11 as $11 \times 11 > 121$

C program

```
#include<stdio.h>
int main()
{
  unsigned int guess, factor, limit;
  printf ("Find␣primes␣up␣to␣:␣" );
  scanf("%u", &limit);
  printf ("2\n");
  printf ("3\n");
  guess = 5;
  while ( guess <= limit )
    {
      factor = 3;
      while ( factor*factor< guess && guess%factor != 0)
        factor += 2;
      if (guess%factor!=0) printf ("%u\n", guess);
      guess += 2;
    }
  return 0;
}
```

Adapted from http://www.drpaulcarter.com/pcasm/

NASM program

Adapted from http://www.drpaulcarter.com/pcasm/

```
%include "asm_io.inc"
segment .data
Message         db      "Find primes up to: ", 0
segment .bss
Limit           resd    1       ; find primes up to this limit
Guess           resd    1       ; the current guess for prime
segment .text
        global  asm_main
asm_main:
        enter   0,0
        pusha
        mov     eax,  Message
        call    print_string
        call    read_int        ; scanf("%u", & limit );
        mov     [Limit], eax
        mov     eax, 2          ; printf("2\n");
        call    print_int
        call    print_nl
        mov     eax, 3          ; printf("3\n");
        call    print_int
        call    print_nl
```

```
        mov     dword [Guess], 5 ; Guess = 5;
while_limit:                     ; while ( Guess <= Limit )
        mov     eax,[Guess]
        cmp     eax, [Limit]
        jnbe    end_while_limit ; use jnbe since numbers are unsigned
        mov     ebx, 3           ; ebx is factor = 3;
while_factor:
        mov     eax,ebx          ; contains factor
        mul     eax              ; edx:eax = eax*eax
        jo      end_while_factor ; if answer won't fit in eax alone
        cmp     eax, [Guess]
        jnb     end_while_factor ; if !(factor*factor < guess)
        mov     eax,[Guess]
        mov     edx,0
        div     ebx              ; edx = edx:eax % ebx
        cmp     edx, 0
        je      end_while_factor ; if !(guess % factor != 0)
        add     ebx,2            ; factor += 2;
        jmp     while_factor
end_while_factor:
        je      end_if           ; if !(guess % factor != 0)
```

```
        mov     eax,[Guess]     ; printf("%u\n")
        call    print_int
        call    print_nl
end_if:
        mov     eax,[Guess]
        add     eax, 2
        mov     [Guess], eax    ; guess += 2
        jmp     while_limit
end_while_limit:
        popa
        mov     eax, 0            ; return back to C
        leave
        ret
```