EuroForth 2020 "Rome"

Preparing for 64 bit

(Praeparatio ad LXIV frenos)

```
-1.0 0 UD.R
340282366920938463463374607431768211446 ok
```

Preparing for 64 bit

Why 64 bit anyway?

Numerical accuracy? 

Addressing range? 

Interface with:
a) Operating system
b) Libraries 

Preparing for 64 bit

Previous experience?

16 bit to 32 bit

FIG-like 16 bit

MPE Forth on WIN32S

1993

Can't remember a thing about it!

Preparing for 64 bit

Critical differences

1

A CELL is no longer an int

Int is still 32 bits

Therefore, @ and ! don't work with ints any more

Preparing for 64 bit

Critical differences

2

Careless Extern: declarations don't work any more

e.g. Enumerations are not ints

Preparing for 64 bit

Critical differences

3

Anything ending with a _t

The good news is, the Linux 2038 problem goes away!

Preparing for 64 bit

Solutions

1

A bit radical, this…..

Get rid of @ and !

a) Use VALUEs instead of VARIABLEs

b) Fetch / store ints in structures using L@ and L!

Preparing for 64 bit

Solutions

1

Even more radical...

Get rid of fetch & store completely

Access structure elements like VALUEs

See my main paper

Preparing for 64 bit

Solutions

2

a) Go through every Extern: and ensure prototypes match exactly

b) Go through every type definition and ensure size is correct

Preparing for 64 bit

Solutions

3

Check every access to to things that end in _t

Preparing for 64 bit

Did it work?

Bit early to tell!