

# About Mobility

## Flash forward

*Hans Hagen*

Mobility is a central issue nowadays. We see the appearance of very small devices that are in fact personal computers. During the 7th ConT<sub>E</sub>Xt-meeting in Bregjlov there were a couple of such devices and after successful installation of T<sub>E</sub>X they were tested on suitability for processing text-files. Although a laptop or a desktop machine is superior it is quite acceptable to use such a device while traveling. It will be interesting to see next developments.

### 1. Introduction

At the 2013 ConT<sub>E</sub>Xt meeting in Bregjlov, Harald König has taken some of his gadgets with him and this time the target was to get ConT<sub>E</sub>Xt running on small devices, most noticeably a mobile phone. You may wonder what purpose this serves, but with such devices becoming more powerful each year, and desktops and laptops getting less popular, we might see the small devices taking their place. Especially when we can dock them in a cradle and connect them to a proper monitor and keyboard we might end up with universal devices. Combine that with projection on our retinas and less tactile input and it will be clear that we should at least look into this from the perspective of T<sub>E</sub>X usage.

### 2. The tests

We used five tests for measuring basic performance. Of course we made sure that binaries and resources were cached.

Test 1 measures some basics, like typesetting a paragraph, flushing pages and loading a file. Because we do lots of pages we can also see if garbage collection is a problem.

```
\starttext
  \dorecurse{1000}{\input ward \par}
\stoptext
```

A normal ConT<sub>E</sub>Xt run is triggered with:

```
context speed-1
```

but with

```
context --timing speed-1
```

memory consumption is measured and one can generate a visual representation of this afterwards.

```
context --extra=timing speed-1
```

We do not show them here, simply because we saw nothing exciting in the ones for these tests. The second test is rather stupid but gives an indication of how efficient the base pagebuilder is:

```
\starttext
  \dorecurse{1000}{test \page}
\stoptext
```

The numbers are normally 10 to 20 times more impressive than those for regular runs.

## contextgroup > context meeting 2015

Test three is a variation on test one but this time we avoid the file being read in many times, so we inline ward.tex. We also add no page breaks so we get less pages but with more content.

```
\starttext
\dorecurse{1000}{
  The Earth, as a habitat for
  animal life, is in old age and
  has a fatal illness. Several, in
  fact. It would be happening
  whether humans had ever evolved
  or not. But our presence is
  like the effect of an old|-|age
  patient who smokes many packs
  of cigarettes per day |=| and we
  humans are the cigarettes.
  \par
}
\stoptext
```

The fourth test draws a few MetaPost graphics, which themselves use a bit of typeset text.

```
\starttext

\dorecurse{10} {
  \startMPcode
    draw fullcircle scaled 1cm
      withpen pencircle scaled 1mm ;
    draw texttext("X") ;
  \stopMPcode
}

\stoptext
```

The last test, number five, is more demanding. Here we use some colors (which stresses the backend) and a dynamic switch to smallcaps, which puts a bit of a burden on the OPENTYPE handler.

```
\setupbodyfont[pagella]

\starttext

\dorecurse {100} {
  \input ward \par
  \dorecurse{100} {
    \dontleavehmode
    {\green this is green}
    {\red \smallcaps this is red}
    {\blue \bf this is blue}
  }
  \par
}

\stoptext
```

### 3. Regular laptops

We started measuring on Haralds laptop, a Lenovo X201i, and got the following timings (that matched our expectations). The second column shows the runtime, the last column the pages per second.

<b>speed-1</b>	5.8	17.1
<b>speed-2</b>	3.6	275.6
<b>speed-3</b>	5.1	19.8
<b>speed-4</b>	0.6	1.8
<b>speed-5</b>	11.9	10.6

Just for comparison, as I'm wrapping this up in 2016, on my current Dell 7600 I get these timings (the last two columns are with LUAJIT $\TeX$ ):

<b>speed-1</b>	4.6	21.9	3.0	33.5
<b>speed-2</b>	3.6	278.2	2.8	357.7
<b>speed-3</b>	4.2	23.6	2,7	37.0
<b>speed-4</b>	0.8	1.3	0.6	1.7
<b>speed-5</b>	6.2	20.3	4.0	31.9

These tests were run with a Lua $\TeX$  0.98 and the most recent Con $\TeX$ t OPENTYPE font processor. As we do more in Lua that a few years back, one can't expect a much faster run, even

when the Dell has a faster processor than the Lenovo. However, what gets noticed is that the fifth speed test runs about twice as fast which is mostly due to improvements in the handling of OPENTYPE features.

#### 4. The Nexus IV

This mobile phone has a quad-core arm processor running at 1.5 GHz. With 2 Gb memory this should be sufficient for running T<sub>E</sub>X. The operating system is Android, which means that some effort is needed to put T<sub>E</sub>X with its resources on the internal flash disk. Access was remote from a laptop.

<b>speed-1</b>	41.9	2.4
<b>speed-2</b>	27.5	36.4
<b>speed-3</b>	38.7	2.6
<b>speed-4</b>	3.4	3.0
<b>speed-5</b>	87.9	1.4

So it looks like the phone runs these tests about five times slower than the laptop. The fifth test is most stressful on the hardware but as noted, a more recent ConT<sub>E</sub>Xt will give better times there due to improvements in feature processing.

#### 5. The Raspberry Pi

The Pi (we're talking of the first model here) has an extension bus and can be used to control whatever device, it has more the properties (and build) of a media player and indeed there are dedicated installations for that. But as this popular small device can host any LINUX distribution this is what was done. The distribution of choice was OpenSuse. The setup was really experimental with an unboxed Pi, an unframed LCD panel, a keyboard and mouse, a power supply and some wires to connect this all. With an ethernet cable running directly to the router a distribution could be fetched and installed.

This device has a single core arm processor running at 700 Mhz with half a gigabyte of memory. Persistent memory is a flash card, not that fast but acceptable. The maximum read speed was some 20 MB per second. It was no real surprise

that the set of tests ran much slower than on the phone.

It took a bit of experimenting but a 200 Mhz overclock of the CPU combined with overclocked memory made performance jump up. In fact, we got a speed that we could somehow relate to the phone that has a more modern CPU and runs at 1.5 times that speed.

Being a regular LINUX setup, installation was more straightforward than on the phone but of course it took a while before all was in place. The default clock timings are:

<b>speed-1</b>	95.841	1.043
<b>speed-2</b>	76.817	13.018
<b>speed-3</b>	84.890	1.178
<b>speed-4</b>	13.241	0.076
<b>speed-5</b>	192.288	0.660

Again, the main conclusion here is that documents that need lots of OPENTYPE feature juggling, this is not the best platform.

#### 6. Summary

We see small devices gaining more performance each iteration than larger machines. Their screens and input method also evolve at a higher speed. The question is if arm will keep dominating this segment, but at least it is clear that they are useable for T<sub>E</sub>X processing. Keep in mind that we used LuaT<sub>E</sub>X, which means that we also have Lua with its garbage collector. Add ConT<sub>E</sub>Xt to that, which is not that small and preloads quite some resources, and it will be clear that these devices actually perform quite well, given slower memory, slower disks, small caches etc. With down-scaled intel chips showing up it can only get better. Keep in mind that we only need one core, so the speed of one core matters more than having multiple cores available, although the other cores can be wasted on keeping up with your social demands on such a device in parallel with the T<sub>E</sub>X run.

A runtime five to ten times slower than a decent laptop is not something that we look forward to in a production environment, but when you're on the road it is quite okay, especially if it can replace a somewhat heavy portable workstation

## contextgroup > context meeting 2015

like we do. Okay, how much T<sub>E</sub>X processing do you need when mobile, but still. As vendors of server hardware are looking into high density servers with lots of small fast processors, we might at some point actually use T<sub>E</sub>X on such hardware. By then performance might be en

par with virtual machines running on average loaded machines.

We are pretty sure that on following ConT<sub>E</sub>Xt meetings more such experiments will be done so we'll keep you posted.