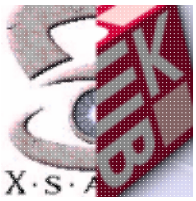
A toolchain for transformation from paper to HTML

by Iznogood
<iznogood/at/iznogood−factory.org>

About the author:

Involved in GNU/Linux for a while, I'm now running a Debian system. Despite electronic studies, I've mostly done a translation work for the GNU/Linux community.



Abstract:

Here is a toolchain to transform a traditional paper magazine into HTML. I will explain the process form the scanning until htmlization.

Introduction

I have read that some US universities will help or allow Google to digitize their library into numeric form. I'm not Google and I haven't such an university library but I've got some old paper magazine about electronics. And the paper quality wasn't the best: Pages start to fall out, the paper become gray...
Then I decide to digitize it because despite the issues stopped about 10 years ago, some articles are always up to date!

Hardware

To begin, I needed to feed the data into the computer. A scanner allows me to do it: after some compatibility check, I bought one, a old used but cheap ScanJet 4300C, and with some internet navigation, I found the needed settings to configure it.
On Debian, I installed sane, xsane, gocr and gtk−ocr as usual with:

```
apt-get install sane xsane gocr gtk-ocr
```

as root.

Sane and xsane are the scanner tools needed by my HP to work.
Gocr and gtk−ocr are tools to make an image transformed into a text.

The scanner is a USB scanner:

```
sane-find-scanner
```

then I went to /etc/sane.d/ to edit some files:
in dll.conf, I uncommented

```
hp
niash
```

and I commented out every thing else.

in hp.conf and niash.conf, I wrote:

```
/dev/usb/scanner0
option connect-device
```

and I commented out every thing else.

I modified the group ownership of the device file /dev/usb/scanner with

```
chgrp scanner scanner0
```

and I add iznogood as user to allow me using the scanner without being root:

```
adduser iznogood scanner
```

One reboot, and it was done!

To store images, DVDs burners are cheap enough to do the job, e.g a NEC 3520. I have an old kernel (2.4.18) so, the IDE burner use the SCSI interface:
With modconf, I have loaded ide−scsi

and I added to /etc/lilo.conf:

```
append="hdb=ide-scsi ignore hdb"
```

then

```
 lilo
```

to take it into operation.
In /etc/fstab, I added:

```
/dev/sdc0    /dvdrom     iso9660    user, noauto     0  0
```

Then I changed scd0 group to cdrom

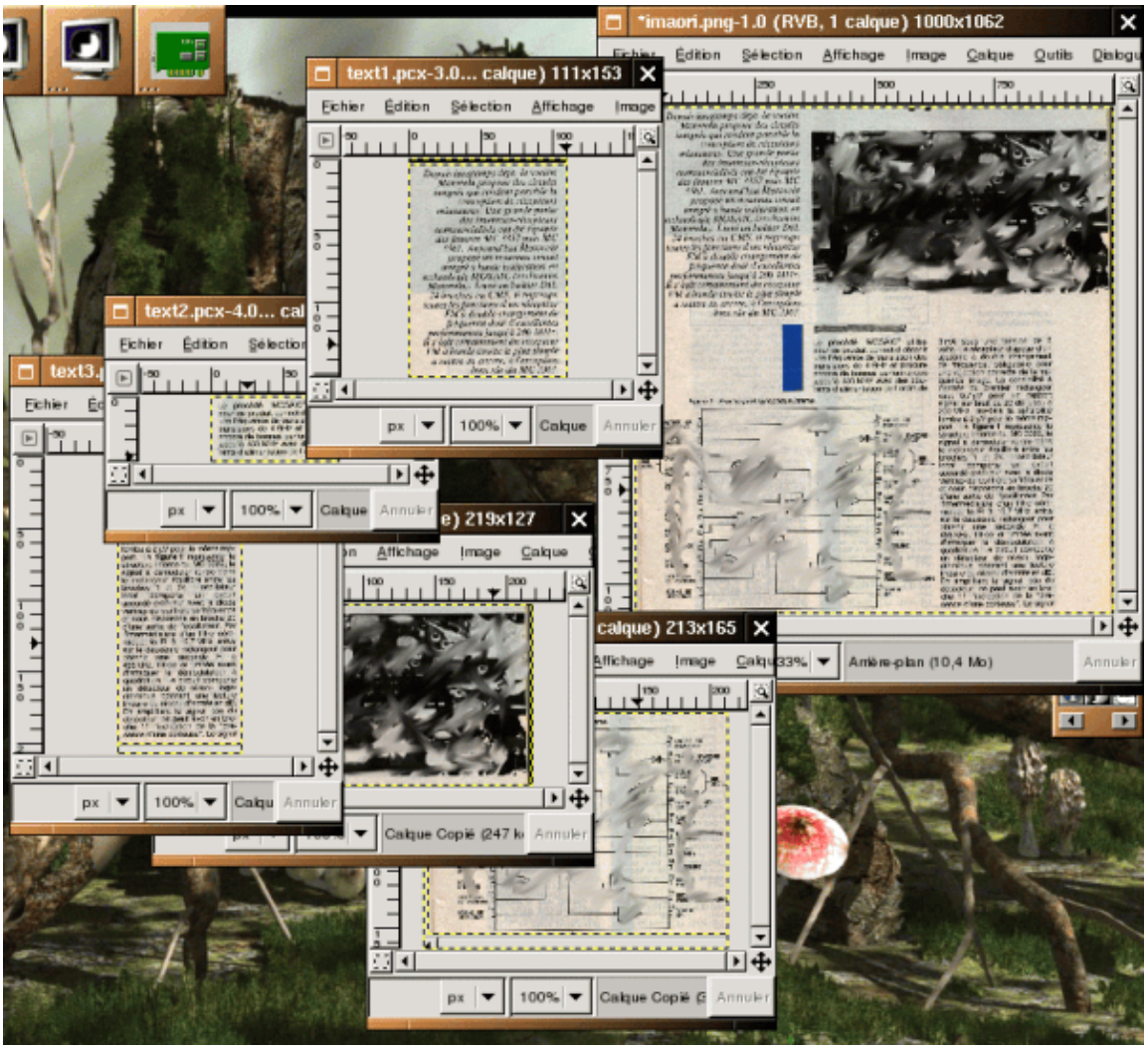
```
chgrp cdrom scd0
```

Quite easy.

Software

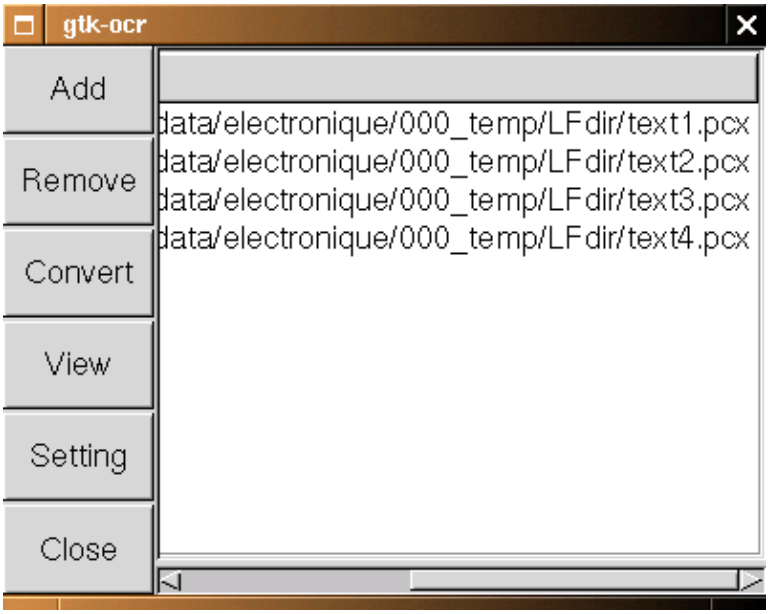
To continue the process, I needed some software:
sane, xsane, gimp, gocr, gtk−ocr, a text editor, a html editor and some disk space.

sane is the scanner backend and xsane is the graphical frontend.
My idea was to keep the maximum resolution and obtain a 50 MB file for one page, store it on a harddisk to work on it and when done, store it on a DVD−ROM.
I put the resolution to 600 dpi, a little bit more brightness and started the conversion. Since it is on a very old machine (a PII 350 MHz), it takes some time but I had a good and precise image. I saved it in png format.
Why such a resolution and a 50 MB file? I wanted to keep a maximum resolution for the archive and for further numeric processing.
Using Gimp I cut the page into graphical images and images containing just scanned in text.
The graphics were saved in png with a reduced size to fit into a html page and the text images weren't reduced but changed from color to gray scale (Tools, Colors Tools, Threshold and Ok) and saved with a .pcx extension for processing with the optical recognition software.



You can see the full scanned image on the top right and the cut parts on the left.
When you are cutting the picture, you can remove titles because they take too much space and they won't be recognized by gocr.
I create a ima subdirectory for images and separate it from .pcx files.

Here comes gtk−ocr, the gocr front end. gocr is a optical character recognition software. It is very simple to use: I just need to select the files and gtk−ocr manages everything. It gave me a .txt file for each .pcx file treated.

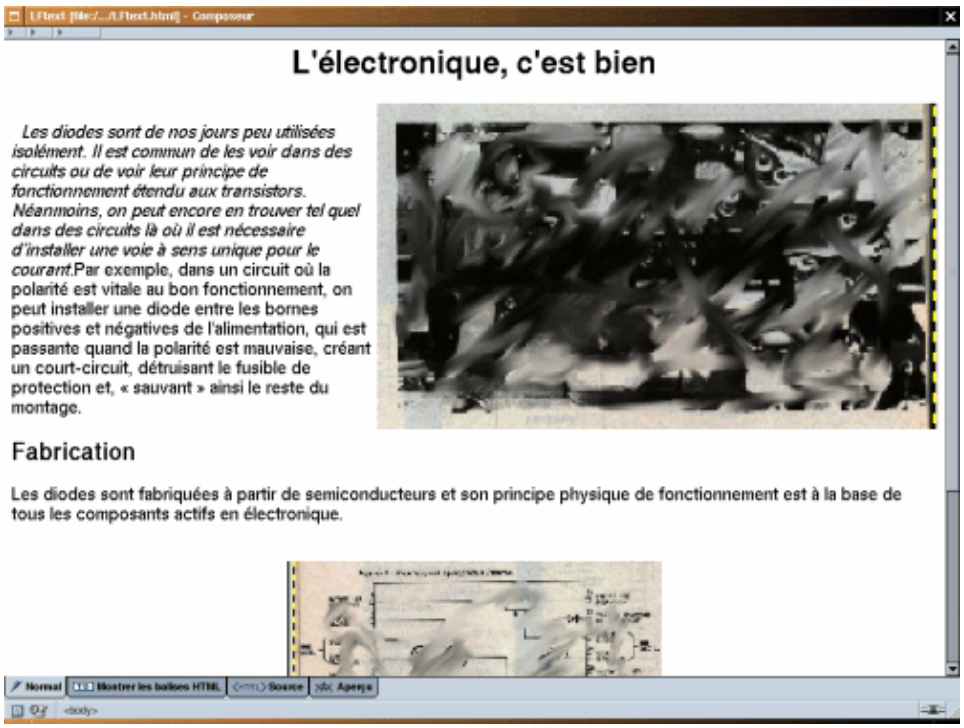


With a simple

```
cat *.txt > test.txt
```

I've got a test.txt and with a text editor I needed to make some adjustments (non french characters removed, words corrected...).

A Copy/Paste to the html editor, Mozilla composer, for me and I started html composition (be careful to have only relative links when you add some pictures).



Bash scripting

I always remember a maths teacher, when I was young, who told me this maxim:

"To be lazy, you need to be intelligent".

Ok, I started to be lazy !!!! ;–)
There is some manual parts which are not easy to automate (directory creation, scanning, gimp cutting and file creation). The rest can be automated.
There is a fabulous English tutorial about bash scripting, ABS (Advanced Bash Scripting Guide), and I found a french translation.
You can find the English version on www.tldp.org.
This guide allowed me to write some little program. Here is the script:

```
#!/bin/bash

REPERTOIRE=$(pwd)
cd $REPERTOIRE
mkdir ../ima
mv *.png ../ima/
for i in `ls *`
do
 gocr -f UTF8 -i $i -o $i.txt
done
cd ..
mv ima/ $REPERTOIRE
cd $REPERTOIRE
cat *.txt | sed -e 's/_//g'  -e 's/(PICTURE)//g' -e 's/ì/i/g' \
-e 's/í/i/g' -e 's/F/r/g' -e 's/î/i/g' > test.txt
```

The file was changed to executable and copied to /usr/local/bin as root with the name ocr–rp.

To make it working, we need to be in the directory to be processed and run:

```
ocr-rp
```

pwd will give the directory path to the script, then ima is created outside the directory and all .png files are moved in. All .txt files are then listed, treated with gocr, concatenated in test.txt and had some changes to fit french characters.

And we continue the same process as before: Copy/paste to Mozilla Composer.
The laziest solution would be to make the script add some headers and footers to the text file, save it and open Mozilla composer directly but I'm too lazy. I will do it tomorrow!!!! ;–)

Conclusion

It was just an overview about digitalization tools and there is, obviously, more than a way to do it and a better one. But there is one constant in GNU/Linux world: the hardware tools are better supported each year and easier to use.

For example, I used a DVD burner to keep my 50 MB images. The installation took me 10 minutes and worked without trouble with k3b (I just had to apt−get install dvdrtools dvd+rwtools).
But with an old PII 350, 192MB RAM, a cheap scanner, DVD burner, some harddrive room, you have a digitalization tool good enough to give "immortality" to an old electronics paper magazine. Here were the tool homepages I used to do the digitalization:

- scanner is a HP ScanJet 4300C
  sane, www.sane−project.org
  xsane, www.xsane.org
- gimp, www.gimp.org
- gocr, gtk−ocr jocr.sourceforge.net
- ABS is on www.tldp.org
- DVD Burner: NEC 3520
- k3b www.k3b.org

2005−06−27, generated by lfparser_pdf version 2.51