

Bagley, Steven R. and Brailsford, David F. and Kernighan, Brian W. (2013) Revisiting a summer vacation: digital restoration and typesetter forensics. In: ACM Symposium on Document Engineering (DocEng '13), 10-13 Sept 2013, Florence, Italy.

Access from the University of Nottingham repository:

<http://eprints.nottingham.ac.uk/28165/1/eprint-202doceng13.pdf>

Copyright and reuse:

The Nottingham ePrints service makes this work by researchers of the University of Nottingham available open access under the following conditions.

This article is made available under the University of Nottingham End User licence and may be reused according to the conditions of the licence. For more details see: http://eprints.nottingham.ac.uk/end_user_agreement.pdf

A note on versions:

The version presented here may differ from the published version or from the version of record. If you wish to cite this item you are advised to consult the publisher's version. Please see the repository url above for details on accessing the published version and note that access may require a subscription.

For more information, please contact eprints@nottingham.ac.uk

Revisiting a Summer Vacation: Digital Restoration and Typesetter Forensics

Steven R. Bagley
School of Computer Science
University of Nottingham
Nottingham NG8 1BB, UK
srb@cs.nott.ac.uk

David F. Brailsford
School of Computer Science
University of Nottingham
Nottingham NG8 1BB, UK
dfb@cs.nott.ac.uk

Brian W. Kernighan
Department of Computer Science
Princeton University
Princeton, NJ 08540, USA
bwk@cs.princeton.edu

ABSTRACT

In 1979 the Computing Science Research Center (“Center 127”) at Bell Laboratories bought a Linotron 202 typesetter from the Mergenthaler company. This was a ‘third generation’ digital machine that used a CRT to image characters onto photographic paper.

The intent was to use existing Linotype fonts and also to develop new ones to exploit the 202’s line-drawing capabilities. Use of the 202 was hindered by Mergenthaler’s refusal to reveal the inner structure and encoding mechanisms of the font files. The particular 202 was further dogged by extreme hardware and software unreliability.

A memorandum describing the experience was written in early 1980 but was deemed to be too “sensitive” to release. The original *troff* input for the memorandum exists and now, more than 30 years later, the memorandum *can* be released. However, the only available record of its visual appearance was a poor-quality scanned photocopy of the original printed version.

This paper details our efforts in rebuilding a faithful retypeset replica of the original memorandum, given that the Linotron 202 disappeared long ago, and that this episode at Bell Labs occurred 5 years before the dawn of PostScript (and later PDF) as *de facto* standards for digital document preservation.

The paper concludes with some lessons for digital archiving policy drawn from this rebuilding exercise.

Categories and Subject Descriptors

D.2.3 [Software Engineering]: Coding Tools and Techniques; I.7.2 [Document and Text Processing]: Document Preparation—*Markup languages*; *Photocomposition / typesetting*

Keywords

Digital restoration, reverse engineering, archiving, *troff*, PostScript fonts, chess fonts, Linotron 202

1. INTRODUCTION

In the 1970s, the Computing Science Research group at Bell Labs, where C and Unix were created, was very active in document preparation research—tools for creating and printing technical documents such as scientific papers and books. That research led to a number of interesting and innovative software tools.

The central component was a program called *troff*, originally written by Joe Ossanna around 1972. *Troff* preprocessors were written for mathematical expressions (*eqn*, by Brian Kernighan and Lorinda Cherry), tables (*tbl*, by Michael Lesk), bibliographic citations (*refer*, again by Lesk), figures and diagrams (*pic*, by Kernighan) and graphs (*grap*, by Jon Bentley and Kernighan).

Troff flourished until the advent of T_EX, and is still used for Unix manual pages (the *man* command uses *nroff*, the typewriter version of *troff*). The suite of *troff* tools is still in use, most often through the modern and polished implementations of *groff*, originally by James Clark; *geqn*, *gtbl*, *gpic* and *grap* are also available.

During the 1970s, the typesetting tools were complementary to some of the other research activities at Bell Labs. For example, in 1974 *eqn* was the first program to use the then-new compiler-compiler *yacc* to implement an unconventional language; *pic* and *grap* also used *yacc* and the *lex* lexical analyzer generator. They were also used to produce high-quality printed documentation like the Unix Programmer’s Manual. Perhaps most important, they were used to typeset technical books, where they helped authors to ensure that complex material was free of errors introduced by copy-editors and printers. Some of those books are still in print, for instance “The C Programming Language”, exactly as they were first created by these tools.

The original typesetting equipment used at Bell Labs was a slow and literally klunky typesetter, the Graphic Systems model C/A/T, or “CAT”. This typesetter, which was intended for small newspapers, produced output on a roll of photographic paper that was advanced a line at a time after being exposed to character images. It had only four simultaneous fonts and 15 sizes. This slow and limited machine served the community well—indeed, its existence spurred the development of *eqn* and *tbl*—but by the late 1970s, it was nearing the end of its useful life. Fortunately better things were on the horizon, with typesetters that created character images digitally on a CRT, not by shining light through a stencil.

The information contained herein is for the use of employees of Bell Laboratories and is not for publication. (See GEI 13.9-1)

Title- Experience with the Mergenthaler Linotron 202
Phototypesetter,
or, How We Spent Our Summer Vacation
Date- January 6, 1980
TM- 80-1270-1,
80-1271-1,
80-1273-1

Other Keywords- reverse engineering

Author	Location	Extension	Charging Case- 39199
Joe Condon	MH 2C-525	6694	Filing Case- 39199-11
Brian Kernighan	MH 2C-518	6021	
Ken Thompson	MH 2C-423	2394	

ABSTRACT

In the summer of 1979, Center 127 purchased a Mergenthaler Linotron 202, a CRT-based digital phototypesetter. This paper discusses our experience with the device, some of what we have learned about how it operates, and the hardware and software we have developed to permit users to take advantage of its capabilities.

Figure 1a: Original page-scan of cover sheet

The group, primarily Brian Kernighan (BWK), spent a lot of time in 1978 exploring new typesetting equipment. The hope was that for a modest price it could get a faster machine that had fewer limits on fonts and sizes, and (a gleam in the eye) might have sufficiently high resolution that it could be used for drawing figures and even for half-tone images.

After much study, an apparently suitable typesetter was found: the newly announced Linotron 202, produced by Mergenthaler, one of the oldest companies in the business. Its likely cost would be about \$50,000, where competing machines were at least twice as expensive, and its specifications implied that it would be much faster than the CAT, far more flexible, and have much higher resolution. Yielding to a modest amount of lobbying, management agreed to the purchase, and the new machine was ordered.

While awaiting delivery, a fair amount of spadework was done, largely by BWK. Ossanna's *troff* was inextricably tied to the idiosyncrasies of the CAT; the number of fonts, the specific character sizes, and many other properties were all wired into the syntax of *troff*, as was detailed knowledge of the intricate device commands to send to the CAT to make it operate.

Clearly this had to be fixed. Unfortunately, Joe Ossanna had died late in 1977, leaving a very powerful but complex and inscrutable program. Accordingly BWK spent a considerable amount of time figuring out (at least approximately) how *troff* worked, and converted it into what came to be known as *ditroff*, for "device independent" *troff*. Many internal limitations were removed, dependencies on the CAT were replaced by parameters, and the output language was converted into a generic format that could be interpreted by drivers for specific devices [1].

Thus when the Linotron 202 was delivered at the beginning of July, 1979, BWK was ready to write a driver for it and move forward. Unfortunately, the 202 turned out to be an operational nightmare. The hardware was flaky and temperamental, Mergenthaler's software was riddled with bugs, the documentation was incomplete and often flat wrong, and even when those problems were temporarily overcome, the machine couldn't be forced into doing what the group wanted it to do.

Eventually all of this was resolved, though only after heroic efforts by Ken Thompson (the creator of Unix) and

The information contained herein is for the use of employees of Bell Laboratories and is not for publication. (See GEI 13.9-3)

Title- Experience with the Mergenthaler Linotron 202
Phototypesetter,
or, How We Spent Our Summer Vacation
Date- January 6, 1980
TM- 80-1270-1,
80-1273-1,
80-1271-1

Other Keywords- reverse engineering

Author	Location	Extension	Charging Case- 39199
Joe Condon	MH 2C-525	6694	Filing Case- 39199-11
Brian Kernighan	MH 2C-518	6021	
Ken Thompson	MH 2C-423	2394	

ABSTRACT

In the summer of 1979, Center 127 purchased a Mergenthaler Linotron 202, a CRT-based digital phototypesetter. This paper discusses our experience with the device, some of what we have learned about how it operates, and the hardware and software we have developed to permit users to take advantage of its capabilities.

Figure 1b: Re-typeset version of cover sheet

Joe Condon (creator of the hardware for the Belle chess machine). The 202 went on to be highly successful for the Bell Labs group, and was used for many years until the advent of high-resolution laser printers and PostScript.

Late in 1979, BWK wrote a description of the work, performed largely by Thompson and Condon, entitled "Experience with the Mergenthaler Linotron 202 Phototypesetter, or, How We Spent Our Summer Vacation." This memo included a long description of all of the hardware and software troubles as reported to Mergenthaler, described superficially how Mergenthaler's proprietary, and deeply secret, character encoding scheme had been reverse-engineered, and explained the new software that had been written.

As might have been anticipated, Bell Labs management at the time was distinctly uneasy about releasing this information, and the memo was suppressed; it was never published externally, and had only limited circulation within Bell Labs.

In a parallel universe in Nottingham, David Brailsford (DFB) was also doing research on document preparation using a Linotron 202, some of which is described in the next section. Since BWK and DFB were well acquainted through their shared interests, various kinds of information flowed back and forth across the Atlantic, including at some point a private copy of the 'vacation memo'. And there matters rested until fairly recently, when DFB decided that it would be of technical interest to re-create the memo with modern technology, in as close to identical form as possible: original fonts, layout, etc., but produced ultimately as PDF. We were further encouraged by a professional typographer and mutual friend, Chuck Bigelow, who, from a history of typography standpoint, also wanted the 'vacation memo' to see the light of day.

This paper describes the re-creation process, and what has been learned along the way. It is suggested that the reader first study the restored original document before reading the following sections, which explain how it was produced. The rebuilt memo is now on the web at [2] and the original scanned version, created from a photocopy of 202 output, is at [3].

As an aside, multiple versions of the original vacation memo existed in early 1980, but because it was suppressed, no decision was made on which version to

release. At this point we have settled on the version for which we have a hard-copy record of its actual appearance; it also has the official Bell Labs ‘cover sheet’ for technical memoranda, and it presents, inline, the original (and we hope entertaining) letter of complaint to Mergenthaler. Inevitably, BWK’s *troff* source text corresponded closely, but not quite identically, to the version we have decided to replicate. There are half a dozen small differences that either correct layout problems and typos or which clarify the exposition.

Figure 1a shows the top half of the cover sheet (page 0) of the vacation memo in the form of a bitmap scan from the photocopied source document. Figure 1b shows the same page area but fully re-typeset using the *mac.scover* macros, where we hope the improvement in quality is evident, even at the reduced size.

2. THE TRANSITION YEARS 1980–1985

Bell Laboratories’ success with the Linotron 202, and the existence of the vacation memo, soon circulated widely in the UNIX community, not least to the parallel universe of the Computer Science Group (CSG) at the University of Nottingham. In the early 1980s CSG was part of a Department of Mathematics but was equipped with its own PDP 11/70 running the UNIX operating system. A mathematics colleague who was in charge of departmental examinations was appalled by the £18,000 annual cost of sending a large number of end-of-year mathematics examination papers for external typesetting. He asked the CSG if a new PDP11 computer, equipped with *troff* and *eqn* software, and driving a suitable external typesetter such as the Linotron 202, might be able to typeset the papers ‘in house,’ thereby reducing costs in the medium term.

To the amazement of all concerned, the University itself agreed to front-up the cost of a LSI 11/23 running UNIX, plus whatever typesetting machine Linotype deemed suitable, and after a period of commissioning in the Department of Mathematics, to move the entire system to the University’s Examinations Unit. One of us (DFB) was appointed as project manager for the first stages of this effort. The University’s longer-term aim was to progress from typesetting mere mathematics towards producing *all* of the University’s examination papers in house.

To their great credit, when approached about this project in late 1982, Linotype UK were quick to admit to the problems that Bell Labs had encountered with the 202. As foreseen in the final paragraph of the vacation memo the Omnitech 2100, although slower than the 202, was seen to be ‘the way forward,’ being both cheaper than the 202 and having the virtue of using laser technology to image directly onto special paper, at a claimed 723 dpi.

The Nottingham team’s trials and tribulations were certainly different from those encountered at Bell Labs, while being every bit as frustrating. Essentially the Omnitech was an early high-resolution laser printer, trying to compete with third-generation film-based typesetters by offering high resolution but without the need for photographic post-processing.

Indeed, by the early 1980s, laser printer technology operating in the region of 300 dpi was found to work very well,

but the push by Mergenthaler to get above 700 dpi required expensive specially-coated paper and very finely divided toner.

The details of the Nottingham team’s adventures are chronicled in [4], where it will soon be seen that if the Bell Labs team had to be armed with screwdrivers to cope with paper jams on the (UK-designed) Linotron 202, then the Nottingham team needed galoshes to wade through seas of toner-ink, caused by leaks in the toner delivery system to the (US-designed) Omnitech 2100’s drum. An alternative version of the Omnitech, using photographic paper or film, was somewhat more satisfactory, though still painfully slow. Eventually, in 1984, the Omnitech 2100 was withdrawn from the market and replaced by the much more reliable Linotron 101.

In a strange twist of fate Nottingham, in late 1983, replaced its trial system of an Omnitech 2100, driven from an LSI 11/23, with a new system consisting of a Linotron 202 (yes !), driven from a PDP 11/44. but still using UNIX and *troff*; Although based on older technology, the 202 positively romped through the work and was a model of sturdiness and reliability. Clearly the four years of extra development on the 202, after the Bell Labs purchase, had done wonders for its robustness.

In the years 1984–87 Nottingham used the 202 for in-house typesetting of all its examination papers. Fortunately this was enough time to recoup the hardware investment because, in 1985, the world of digital documents changed for ever with the advent of the Apple Laserwriter running the Adobe PostScript language. PostScript was designed as a graphics language with a high degree of device-, and resolution-, independence. It implemented the entire range of vector graphics constructs — lines, arcs, splines, etc. — and was able to apply these constructs to the shapes of character glyphs within fonts. Rendering speed was helped by having optimised subroutines for character glyphs within the so-called Type 1 font format, coupled with ingenious ‘hinting’ techniques for preventing pixel rounding problems at low resolution. Indeed, PostScript on the Apple Laserwriter showed that, even at 300 dpi, there was a market for quality typesetting. Soon afterwards the language migrated to the Linotron 100 and 300 series machines and spread rapidly thereafter. A ‘display’ version of PostScript was also developed for on-screen preview of PostScript documents and, by 1989, the victory of PostScript, in the print and publishing industries, was total.

All that was now needed, to complete the PostScript saga, was an ‘interchange’ form of it, optimised for fast rendering, device independence and document exchange. This appeared in 1992 as Portable Document Format (PDF) and it first came with an interpreter called Adobe Acrobat, available initially on Macintosh and PC. In the ensuing 20 years PDF became first a *de facto* standard, and later a full ISO standard, for the interchange of high-quality, print-ready documents of arbitrary complexity.

So, we are now able to make clear the aim of this project, which was simply to rebuild the vacation memo, not just as concatenated low-quality page-scans, but as a typeset-quality PDF file. In this way it could join the various existing PDF archives of Bell Labs memoranda.

Given that the Linotron 202 is long obsolete we needed to consider what combination of software and fonts might best rebuild the vacation memo with good enough quality for readers to get a clear feeling for the 202's capabilities.

3. TOOLS FOR THE REBUILD

In any restoration project there needs to be a degree of continuity between the tools, techniques and materials available at the time the original work was created and those now available at the time of restoration. For example, in a previous restoration project, involving UK Parish Registers [5], the aim was to recreate hard-copy volumes of Derbyshire marriage registers that had gone out of print in the early 20th century. Page scans of these original volumes were available from genealogy web sites, capable of giving reasonable character recognition when fed into OCR software. Recreating the simple tabular layout for the registers was not a problem. The real challenge was to find out whether the fonts used in the original printed registers were still available. The body-text font was readily identified as Caslon (with Old Style figures). The other two fonts used in various headings were accurately identified by *WhatTheFont* as Romana Bold, together with a font in the Gothic style called Fordor Incised. Both of these were initially created in the days of 'hot metal' typesetting but were now available from two different vendors as PostScript fonts. Indeed it is a testament to the availability of tens of thousands of fonts in the digital typesetting era that every bit as much effort seems to have been spent in 'rescuing' old typefaces as in creating new ones.

In the present project things were a little different, because *troff* source code for the memo still existed. But yet again fonts featured large in the restoration effort. The vacation memo needed eight different fonts whose identity was known from the outset. Five of these were readily available but, as we shall see, three fonts had to be recreated from scratch.

The few surviving hard-copies of the vacation memo are simply photocopies of a typeset original, with an appearance very similar to that shown in Figure 1a. On the other hand, because BWK had at least preserved a reasonable approximation to the *troff* source code of what we were trying to replicate, there was no need to resort to OCR for text acquisition. However, the very subject matter of the memo was the exact 'look and feel' achievable on a Linotron 202, an obsolete typesetting machine. Moreover this look and feel had to be replicated as closely as possible in PDF—a format which did not become available until 12 years after the vacation memo was written.

Despite trying hard for an accurate match to the metrics and appearance of the original memo, it was inevitable that small differences in character widths would accumulate. This in turn might cause *troff* to make different line-breaking decisions in the rebuilt version compared to the original. For this reason it was decided that different line breaks could be tolerated in the rebuild, but page breaks would be kept as near identical as possible. This would ensure that the rebuilt memo, like the original, would occupy 14 pages (including the cover sheet), with the main body of the paper ending on page 11. This being said, we

took the opportunity of adjusting one or two page breaks to achieve better formatting. A particularly clear example is the 'widow' at the top of page 8 in the page-scan version of the original memo, which has now been taken back onto page 7 in the rebuilt version.

3.1. Software availability

The vacation memo makes clear that the imminent arrival of the Linotron 202 was the spur for BWK to develop *ditroff*. The 202's line drawing capability (nothing more than printing large numbers of dots very close together) also prompted the development of the *pic* language for creating line diagrams [7]. Thus the first step in recreating the vacation memo was to find a version of *ditroff* and its accompanying macros that was ancient enough to cope with 1979-vintage *troff* source code. Fortunately DFB has resolutely used, and maintained, a version of *ditroff* that is of mid-1980s vintage. It still has traditional two-letter names for *troff* fonts, plus the original hyphenation algorithm that pre-dates the introduction of T_EX hyphenation in the 1990 version of *ditroff*.

But this still left the problem of locating a corresponding version of the standard *troff ms* macros and, in particular, the original Bell Labs version of the ancillary macros called *tmac.scover*, which control the appearance of cover sheets for Bell Labs internal memoranda.

As part of the release of UNIX System V in the mid-1980s an extra-cost package called Documenters' Workbench (DWB) became available, gathering together *ditroff* and a host of other post-processors and pre-processors.

In a wonderful demonstration of the web as a 'crowd-sourcing' repository, a download of Version 3.3 of DWB (dated 1992 and now free of charge) from the Bell Labs web site revealed that, in addition to DWB's own *mm* macros, there was also a carefully preserved version of the original *ms* macros, including the all-important Bell Labs internal version of *tmac.scover*.

An early triumph was the processing of the *troff* source text for the cover sheet, which after a few minor adjustments gave the output seen in Fig 1(b).

The processing pipeline for the remainder of the vacation memo, under Open SuSe Linux, soon settled down as:

```
pic 202paper.trf | psroff -ms -t | distill16 > 202paper.pdf
```

Here *psroff* is a shell script that invokes *ditroff* and then feeds its output into a PostScript back-end provided by Adobe TransScript 4.0. The PostScript output from this back-end is then converted into PDF by being piped into a Linux-compatible version of Adobe's Distiller 6.0 software. Note that the *-ms* flag, in this case, denotes access to a mid-1980s version of the macro set, together with all the subsidiary macros for cover sheets, etc.

Just as when the vacation memo was first created, the only *troff* preprocessor needed is *pic*. Its first challenge was to handle the code for a line diagram on page 7 of the vacation memo. This drawing shows the component vectors of the Helvetica letter 'e' in the Linotron 202 font representation; it is shown again here in Figure 2.

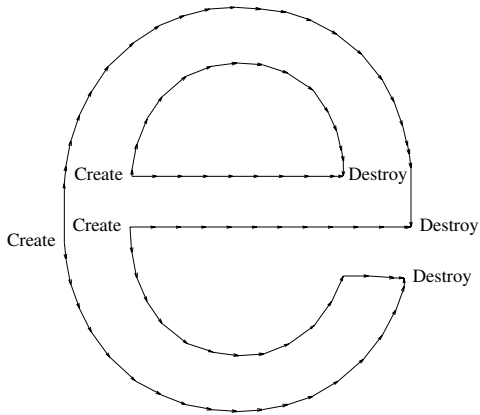


Figure 2: The Linotron 202 line vectors for Helvetica ‘e’.

3.2. Font availability

In this section we look at the eight fonts needed for the reconstruction effort. Ideally, for total authenticity, the font recreation should have started from the original, line-segment, font outlines used by the 202 itself (see Figure 2) and using all the knowledge from the 1979 ‘vacation project’ of exactly how these outlines were represented. With some effort the line-segment data could have been converted into Adobe Type 3 (unhinted) PostScript fonts. Two factors ruled out this approach: firstly much of the original Linotron 202 font data appears not to have been archived, either at Bell Labs or at Mergenthaler-Linotype, and secondly the emergence of PostScript Type 1 versions, for six of the eight fonts needed, from Linotype and Adobe, gave a readily available set of PostScript successor fonts.

Once the PostScript revolution got under way in 1985, it was just a matter of time before this language migrated from laser-printers onto higher resolution typesetters. The first such high-resolution typesetting machines to offer PostScript were the Linotron 100 and 300 series. A cross-licensing deal was signed for Adobe’s PostScript and Linotype’s fonts in that same era. It followed naturally therefore, that a large number of typefaces in the Linotype font catalogue were eventually re-implemented as PostScript Type 1 fonts, created and distributed by Adobe.

For this reason the Adobe Times typeface family was some sort of clear successor to the Linotype Times family used originally on the 202. Moreover the other fonts mentioned on page 8 of the vacation memo (in a table illustrating average number of bytes per character in the 202 encoding scheme) were also readily available for the rebuild project. For reference this table was:

Helvetica	91
Memphis Medium	104
Times Roman	113
Old Englishj	168

Here, Times and Helvetica are standard fonts, supplied with just about any PostScript-based typesetter. Memphis was also readily available as an Adobe font while Old English, despite being supplied by Linotype for use on the 202, is actually a Monotype font, and hence was not

PostScript-converted by Adobe. Nevertheless, it proved to be readily available as a PostScript font from the same FontShop (in Cheltenham UK) that used to supply fonts for the entire Linotron range.

Of the three fonts that had to be reconstructed from scratch, two of them—Print Out and ChessKLT—were sufficiently problematic as to merit entire sections to themselves, later on in this paper. All that remains, for the moment, is to devote brief sub-sections to the questions of the Courier font and the Bell Labs logo and its variants.

3.2.1. The Courier typeface

The monospaced Courier typeface was commissioned by IBM and designed by Howard Kettler in 1955. IBM’s decision not to copyright or trademark it has led, over the years, to numerous variants being available. The font aims to replicate the effect of a typewriter striking through a ribbon onto a sheet of paper and Adobe’s PostScript Type 1 Courier is proportioned to set 10 characters per inch at a 12-point body size. It remains a mystery as to why Linotype did not make Courier available for its typesetting machines (though many other proportionally-spaced typewriter faces *were* available).

Despite the availability, on the 202, of a fixed-pitch font called Print Out, which was given the *troff* name of PO (see later section), the programmers at Bell Labs still felt the need for the Courier font. Page 8 of the vacation memo relates how artwork for Courier was taken from the second-generation CAT typesetter and was then scanned and converted for use on the 202 and given the ‘traditional’ *troff* constant-width font name of *cw*. A sample of this font is shown near the top of page 11 of the memo, where it will also be noticed that several non-ASCII symbols (e.g., arrows, open box and cent) were present. In deciding how to replicate this font for the present rebuild project, some other considerations needed to be addressed.

From the earliest days of PostScript typesetting, Adobe realised that the Courier family needed to be a component of the ‘Adobe 13’ standard fonts, available with all PostScript typesetters, since it had achieved a status of a *de facto* standard for tasks such as program printouts.

Unfortunately, it has been a considerable source of frustration, to many users over the years, that Adobe’s Type 1 version of Courier, and variants such as Courier New, are now rather too ‘light’ in colour to harmonise readily with typeface families such as Times Roman.

For this reason it was decided to use an online, publicly available, Courier Dark from Hewlett-Packard to emulate the appearance of the CAT constant-width font on the 202. The download was initially available as a TrueType font for PC use, but after format conversion in FontForge it was exported as an Adobe encoded Type 1 font.

Figure 3 shows a bitmap scan of the CW font, extracted from page scans of the vacation memo, alongside the type-set reconstructed version based on Courier Dark.

3.2.2. The Bell Logo font

From the earliest years of the second-generation CAT typesetter, Bell Labs always made sure that the Bell

```

a b c d e f g h i j k l m n o p q r s t u v w x y z
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
0 1 2 3 4 5 6 7 8 9 & . , ; ? ! ( ) - ' ` ~ $ % ^ _
/ ' [ ] * " # ~ @ + - ! \ > < = ^ _ { } | ! ~ - + < ^

```

Figure 3a: Bitmap scan of GSI-CAT CW font

System logo was present within its Special Font, which consisted of mathematical and other symbols. UNIX users outside of Bell Labs just had to accept that any callout of the Bell logo, via `\(bs`, would absolutely *not* deliver the Bell logo but, in all probability, something like ♥.

In order to recreate the Bell logo as a font character, the first requirement was high-quality artwork. Once again the web in general, and Google Images in particular, came up with the goods. A 27 Kbyte drawing was found of sufficiently high resolution (well in excess of the 1000 × 1000 units resolution of a Type 1 font glyph) that the Adobe Streamline program was able to fit it with a very accurate outline. Once the outline was imported into Adobe Illustrator it was easy to export it as an Encapsulated PostScript .eps file.

The Bell logo ☉ is required (at various point sizes) for the top of the vacation memo cover sheet, the top right of page 1 of the memorandum and also as an in-line insert near the foot of page 1. Initial tests were performed with the *psfig* program [8] for effecting inserts of PostScript material into *ditroff* source code. Once this was working correctly Adobe Illustrator was used, once again, to create the ‘side-ways’ version ☉ which appears just over half way down page 9 of the vacation memo. The visual effect was achieved by selecting just the inner ‘bell’ element within the outer circle of the logo, and rotating it clockwise by 45 degrees. After further testing with *psfig* the two versions of the logo were imported into a special two-character font called `BL`, created under Fontographer.

4. REBUILDING THE Print Out (PO) FONT

For reasons we have been unable to ascertain, Linotype did not make Courier available as its fixed-pitch typeface family for the Linotron typesetters. Instead, they bundled in a fixed-pitch font called Print Out which had upright and bold versions. Print Out can be seen in action near the foot of page 9 of the vacation memo, where a sample of it is displayed. It is also appears as the chosen font for BWK’s letter of complaint to Mergenthaler which stretches from page 3 to page 6 of the memo.

Extensive enquiries in England, Germany and the US failed to find a copy of it and the news, from what remains of the Linotype organisation[†], is that Print Out was never converted for PostScript use. It’s not too difficult to discern the reason: Print Out is not exactly an elegant thoroughbred. This font would have to be completely re-drawn if a PostScript version could not be found,

[†] Since 2007 Linotype has become a subsidiary of Monotype. This may well reflect the harsh new realities of the digital fonts era, but it still feels akin to Ford being taken over by General Motors.

```

a b c d e f g h i j k l m n o p q r s t u v w x y z
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
0 1 2 3 4 5 6 7 8 9 & . , ; ? ! ( ) - ' ` ~ $ % ^ _
/ ' [ ] * " # ~ @ + - ! \ > < = ^ _ { } | ! ~ - + < ^

```

Figure 3b: Re-typeset version using Courier Dark

Even so, recreating this font, while not difficult, took several tens of hours of painstaking effort. Fortunately DFB had been given a Linotype Font Shop catalogue in 1983. This catalogue had a high-quality sample of Print Out, very clearly the product from a professional printing press and imaged onto good-quality glossy paper. This sample was certainly a little larger in point size than the examples available in the memo but how much larger?

We already know that standard PostScript Type 1 Courier is proportioned to set 10 characters per inch at a 12 point body size. This equates to 1/10 of an inch in character width or 7.2 points (assuming we adopt the ‘Adobe point’ which has 72 points per inch). Now, $7.2/12 = 0.6$ which means that the width of Adobe Courier is 600 units on the standard 1000-unit em square used in Type 1 fonts.

However, many typewriter-derived typefaces are proportioned so as to mimic pica-sized (12 point) type. Here the basic proportions of the glyphs will be 1.2 times wider than the 600 em-units that define the 10 point designs. Given that $600 \times 1.2 = 720$, it follows that fixed-width designs for such typefaces will set only $10 \times 600 / 720 = 8.33$ characters per inch at 12 point. Finally, the above calculations imply that if the 26 upper-case alphabetic characters of a fixed-pitch font are typeset in sequence, with no spacing between them, then the total width will be $26 / 8.33 = 3.12$ inches.

The first three lines of the Printout sample in the font catalogue were as shown in Figure 4a. Moreover, the width, in the catalogue, of the upper-case letters on the second line of Figure 4a was 3.1 inches: the sample was indeed indicative of a typeface with a native body sizing of 12 point and a set width of 720 units.

The font-catalogue sample for the PO font was scanned at 600 dpi and Photoshop was then used to separate out each glyph and export it as a .tif file. These glyphs were then read in to Adobe Streamline to fit curves around them. The resulting outlines were exported as .eps outlines.

At this stage it became urgent to identify the provenance of the Print Out font. Chuck Bigelow pointed out that it had close similarities to proportionally-spaced typefaces, such as Corona and Century Schoolbook. A small clue led to the final identification—the ‘ear’ at the upper right of Print Out Roman’s lower-case g looked identical to that on the same character in Excelsior Roman. Excelsior is a precursor of Corona and both of these typefaces had been PostScript-converted by Adobe in the mid-1980s.

4.1. Creating the basic shapes for Print Out

The way ahead was now daunting, but very clear. Each of the fitted outlines for the scanned Print Out glyphs was imported into Fontographer, as a template background layer for a new Print Out font.

abcdefghijklmnopqrstuvwxy
ABCDEFGHIJKLMNopqrstuvwxyz
1234567890

Figure 4a: Sample of scanned PO font at 600 dpi

Then one by one, the corresponding glyph outlines were copied over from Excelsior to act as a starting point for the foreground layer. The character widths of the alphabetic glyphs, in Excelsior, range from 333 for ‘i’ to 1000 for ‘W’. But now all of these shapes have to be coerced into a fixed-pitch 720-unit width and re-moulded to match the scanned-in outline in the template layer. For some glyphs (such as W, M, w, m and n) major surgery was necessary to reduce or remove prominent serifs, followed by compression of the stem spacings. By contrast, narrow letters such as i and l needed serifs to be extended and the crowning dots of i and j needed lowering and enlarging. Varying degrees of stretching, shrinking and minor surgery were also necessary on the bowls, loops, tails, crossbars and counters for characters such as b, d, f, g, j, p, q, t, and y. Since the Excelsior digits 0-9 were already designed at a fixed pitch of 556 units, relatively little stretching and adjustment was needed to adapt them for a 720-unit set width. Adjustments to the various bracket and punctuation glyphs were easy and relatively minor, usually amounting to little more than stretching of stems and minor adjustments to stem weights.

In all of these adjustments a close watch needs to be kept on x-heights of lower-case letters; the PostScript Type 1 version of Lucida Typewriter, from Bigelow and Holmes, was used as a guide. This font is also a fixed-pitch, pica-size, 12 pt design with an x-height set at 530 em-units. This x-height seemed to correspond well with the 12 pt x-heights for PO seen in Figure 4a. However, the non-linear nature of human vision means that this is only the start of the story; a host of small height adjustments has to be made to lower-case letters to make them ‘look right.’

4.2. Sidebearings for Print Out glyphs

The final task in creating the PO font was to adjust the left and right sidebearings for each character. To illustrate, let us consider the letter o in the PO font. Within the set width of 720 em units the glyph itself occupies 594 units. This leaves 126 units for the sidebearings, i.e., the space before and after the glyph itself. Other things being equal the spacing of a fixed-pitch font is never going to look as elegant as a proportionally-spaced one. However, the worst of the visual effects can be mitigated a little by moving glyphs very slightly left or right within the fixed-width character cell.

So, as a first approximation to getting things right, the spare space for sidebearings, in each Print Out character, was allocated, left and right, in the same ratio as in Lucida Typewriter. Thereafter multiple further sidebearing adjustments have to be done, firstly to harmonise the way that single-stem and multi-stem characters appear in conjunction with the letter o. Having tried strings such as

abcdefghijklmnopqrstuvwxy
ABCDEFGHIJKLMNopqrstuvwxyz
1234567890

Figure 4b: Re-typeset sample from the new PO font

nonono and uououo one then progresses to doing all the lower-case characters, against o, in turn. Once this has been completed one can move on to harmonising strings of the most frequently occurring digraphs in the English language such as thththth, hehehehe, anananan, inininin, and so on.

As can readily be imagined, the whole effort is a time-consuming compromise. A tweak of sidebearings that makes one digraph look pleasing will almost certainly make some other digraph look awkward.

Examples of original and rebuilt Print Out can be seen in Figure 4 and also, at greater length, in the original and rebuilt versions of the vacation memo.

5. APPENDIX DIAGRAMS

The Appendix to the vacation memo was written by Joe Condon. In it he presents diagrams showing the detailed nature of the parallel interface between the PDP11 and the 202. These interconnection and logic diagrams were created in UNIX *plot* format. A converter called *pltroff* had been written by BWK to map *plot* codings into *pic*, but initially we could find no trace of the C source code for it.

Thus we decided to recreate the *pic* for figures A1-A3 from scratch, since DFB already had an extensive library of *pic* shapes suitable for the logic gates in these figures. We also recreated the ‘ff1’ box in Figure A2 (missing from page A-2 of the page-scan original). The smaller body size of the Courier (CW) font made the numberings on the diagrams much more legible than they were on the original page scans, where PO had been used.

We later found source code for *pltroff* in FreakNet’s Media Lab [9], but decided to retain the hand-optimised *pic* diagrams in the rebuilt Appendix.

6. RE-CREATING THE CHESS FONT

As more and more of the rebuilt memo attained the typeset quality we were seeking, the one object that increasingly cried out for attention was the diagram on page 9 of the memo, showing Ken Thompson’s chess font in action.

Thompson himself (KLT) was contacted to find out if he could help. His reply [10] initially held out hope that he might be able to locate the original 202 chess font in his archives. Sadly this has not materialised, but he revealed that the artwork for his chess pieces came from *Chess Life*, where the the pieces were logo headings to the different sections in the magazine. The scanning and font construction was done, in a hurry, to illustrate a series of books by David Levy (an International Chess Master).

The helpful thing about KLT’s original font is that, given the need to produce something quickly, the chosen outlines were simple and seemingly based on drawings that were

one inch square with a grid resolution of 0.05 inch. This simplicity meant that it was easy to create mock-ups of the piece shapes using *pic* and these approximate shapes were then handed over to Steve Bagley (SRB) for further development. The hope was that a fully functioning PostScript font could be devised that was reasonably faithful to the 202 original.

In terms of actually reproducing the chess board illustration in the vacation memo, a fascinating insight into what went on in 1979 was given by KLT's *troff* typesetting code for that board position, which was as follows

```
.ft CH
zyaiydyiygyicytez
zikiagbaibz
zbibijqbbsdz
ziaiaiaiaz
zailiaiaiz
ziaslaijiaz
zjqjjiaqjaqjz
zxixoxixmxixaxiaz
```

Now, there was no reason for us, necessarily, to design the font so as to match this typesetting source code. But we thought an analysis of the above instructions would be instructive — and so it proved.

We note, at the very outset, that neither the black nor the white queen shape appears in Figure 5a — the chessboard position seen in the vacation memo. However, all other pieces do occur in Figure 5a and this enables the detective work to begin.

Each of the eight rows of characters in the typeset input starts and finishes with *z*, so it is likely that this letter corresponds to vertical segments of the edging that surrounds the board. Corresponding edging pieces for the top (*y*) and bottom (*x*) boundaries can be seen in lines 1 and 8, where these letters are interleaved with other letters representing the actual chess pieces on that row. Since there are no backspacing motions, *x* and *y* must behave as overstriking, i.e. zero-width, characters. Analysis of line 5 of the *troff* input against the corresponding, unoccupied, row of the board shows that *a* must be a white square and *i* a black (shaded) square. After a little more work we discover that the black pieces (pawn to king) occupy character slots *b–g*. By contrast, the white pieces occupy *j–o*.

This only leaves the problem that, by default, all of these black (or white) pieces will be typeset on a white background. To achieve a black rook on a shaded background (e.g., at the top right of the board at position *h8* in chess notation) the required coding seems to be *te*, which shows that KLT has cunningly superposed a black rook, *e*, shape on top of what must be a shaded background with a rook shape cut out of it, and this has been assigned to the letter *t* of the CH font. A little more analysis then shows that these 'cutout shapes' for the various pieces must occupy positions *q–v* and, like the edging pieces already discussed, must be treated by *troff* as being of zero width. A confirmation of much of the above analysis came, yet again, from the Freaknet repository [9], which yielded the *ditroff* width metrics for the CH font. These metrics confirmed that all of the characters in the ASCII range

a–z were in use and all of the shapes assigned to these positions did indeed have constant width, with the zero-width characters being exactly those we had predicted.

6.1. Shapes of the king and queen pieces

At first sight the white and black kings (at squares *b1* and *e8*, respectively, on the vacation memo chessboard — see Figure 5a) might seem to be completely different designs. In particular the white king seems to be adorned with an inverted black diamond at the very top. However, closer inspection shows that the 'black diamond' effect on the original diagram results from the scaling down of a white cross, accompanied by a generous helping of ink bleed at the various stages of photocopying.

Artwork for the missing queen shape in KLT's chess font — a three-pointed crown — was eventually obtained from the cover, and the interior, of one of the aforementioned books by David Levy [11]. Improved artwork for the other shapes was also obtained from that same source.

6.2. From *pic* shapes to PostScript fonts

The creation of a Type 3, unhinted, replica of the CH font proceeded as follows. The piece shapes created by DFB in *pic* were first exported via *ditroff* and Adobe Distiller to PDF, one to a page, and at a size of 8 inches wide. The programmatic nature of *pic* allowed us to create the shapes easily, but it did throw up some problems of its own. Firstly, *pic* creates outlines, not filled shapes. Secondly, *pic* creates each line, arc, or spline individually, in the order specified by the *pic* programmer. It makes no attempt to create a connected path but such a path is essential for 'fillable' shapes like the black pieces.

An Objective-C program was written that parses the PDF definitions for the pieces, and builds paths from the individual lines produced by *pic*. Each piece is parsed into an array of lines and a set of points is built up containing the start and end points for each line. The algorithm then picks a point from the set and finds all lines that either start or end at that point. Ideally, this enables pairs of lines to be joined together and replaced by a single line, which is then put back into the array. The algorithm continues until no more lines can be joined together. The result is a series of joined lines that all start and end at the same point, which represent the distinct segments of the original shape

For instance, the rook decomposes into two segments: the pedestal and the battlements, while the king has three components. These segments can then be exported as normal PostScript paths (with any curves flattened into straight lines to echo the way that the 202 approximated a curve) and then filled to form the black pieces in the font.

6.3. Black and white pieces

The approach described above gives paths that can be used to form the black pieces. Inspection of KLT's chess characters, see Figure 5a, shows that, almost certainly, he created the white pieces out of the black ones by drawing an exterior outline and then throwing away the black interior. Exactly the same procedure was followed in recreating the CH font: further software was written that took the path for the black pieces and produced a new outline, which was

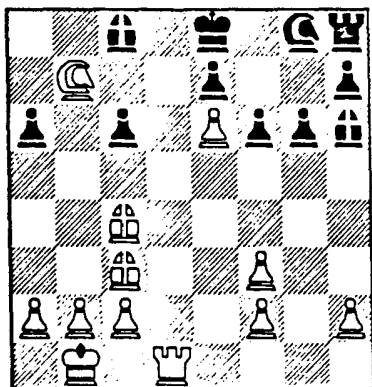


Figure 5a, Original chess diagram from the vacation memo

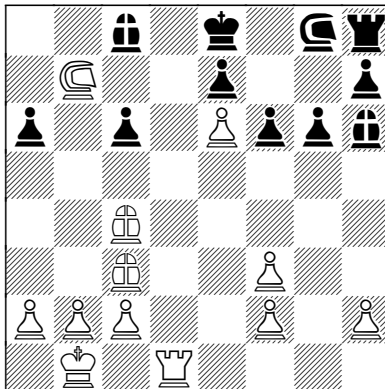


Figure 5b: Same chess position typeset in the CH font.

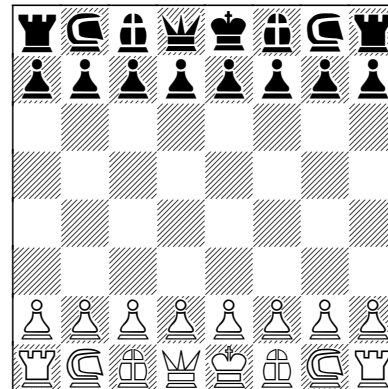


Figure 5c: Starting board, typeset in the CH font.

equivalent to a stroke around the outer edge of the path. This was produced by taking each segment of the path and calculating the position of a new line segment that was parallel, and to the left of, this piece by the desired width of the line. This results in a series of new, but disconnected, line segments. The lines were reconnected by shrinking or extending them until they intersected with the immediately preceding line segment. For this process to work correctly all the paths must be drawn in a clockwise direction and so the paths were pre-processed to impose this condition.

6.4. Creating the cutouts

The final process was to create the “shaded square with piece hole” glyphs described above. This again was performed programmatically by considering the intersections between the piece path and the path representing the hatching lines for the black square. It was realised that the calculations could be simplified if everything was transformed such that the hatch line was running horizontally along the x -axis from the origin. This approach highlighted a number of interesting optical side effects that needed to be mitigated. Firstly, cutting the lines based on the black-piece path data still produced visible collisions since the actual imaged line is wider than the mathematical one. This required the cutter software to use an enlarged path (similar to the mechanism used to create the outline) and also to consider the width of the hatch line.

In essence the process consists of calculating the intersection coordinates of each diagonal shading line with the various segments of the chess piece that is to be superimposed upon them. Once they are calculated the chopped line lengths are shrunk by about 5%, to give a fit that is tight, but not too tight. Problems arise with diagonal lines that very nearly intersect the chess shapes. These near-tangential lines are precisely the ‘optical side effects’ referred to in the previous paragraph.

6.5. Results

Figure 5b shows that a close approximation to KLT’s typeset chessboard diagram can be rebuilt and with a visual quality far better than that available from the page-scanned

version in Figure 5a. As a test of the viability of the rebuilt font, Figure 5c shows a chess-game starting position, typeset from our new font; it also shows the newly discovered shapes for the black and white queens.

7. CONCLUSIONS

The work done on the Linotron 202 in 1979 was influential, though only indirectly. Document preparation was a major area of research for a significant number of computer scientists, and it provided an outlet for innovative work in tools, languages and even mathematics; think of the progression from *tbl* and *eqn* to \TeX , and from simple character outlines to Metafont. And the importance of allowing authors to typeset their own work should not be underestimated; though that was once unusual, it is now the norm for most technical authors.

But the work described in the vacation memo was, in some ways, just a little too early. Deducing how the fonts were encoded is a graphic example of how security by obscurity is ultimately doomed; no matter how well a secret seems to be protected, a sufficiently motivated attacker is likely to find a weak spot. Even if Mergenthaler had been more willing to share its expertise, however, few small research operations could afford an expensive machine for experiments; it was only the promise of production use that made the 202 viable at Nottingham, for example.

Once hardware costs dropped by an order of magnitude with the advent of PostScript and the laser printer (Bell Labs got its first laser printer, from Imagen, around 1982), the field opened up to a great wave of creativity: people with new ideas could put them into practice without having to be font designers and without having to buy expensive machines. Of course not everyone was a skilled font designer; quite the contrary, and the new wave also unleashed a tide of poor-quality fonts and rampant font piracy. But, in the end, quality shows; once Adobe and Linotype began distributing high-quality PostScript fonts these standards became the norm. Of course these fonts helped us greatly with the work on the vacation memo; had it been some non-Mergenthaler typesetter, conversion of the fonts we needed into PostScript format, with the same character metrics, might have been less easy.

PostScript is a fusion of typography, computer graphics and programming language design. The typesetter design community, talented though it is, would not have come up with PostScript. Nor would the computer science community have come up with the rich repertoire of fonts that came from typography. Today, tools like Fontographer enable mere computer scientists to work on fonts like PO, but lasting designs will only come from professionals.

It has been almost as much fun to work on reconstruction as it was to work on the original projects at Bell Labs and Nottingham. But computer archaeology has its problems. To paraphrase George Santayana, “Those who do not archive the past are condemned to recreate it.” During this reconstruction, we have been frequently surprised and often discouraged by how much information has disappeared in 30 years. Most obvious, the details of the Mergenthaler character representation, a very clever and compact technique that was reverse-engineered only with painstaking detective work, seems to have gone completely. The representation was never written down, except implicitly in the *ad hoc* programs that were written to process fonts, and those programs have long since disappeared. Perhaps someone at Mergenthaler-Linotype has the information, and clearly there are analogous and documented mechanisms used by PostScript, but it is unfortunate that this part of history seems to be gone forever.

The fonts that were laboriously constructed to take advantage of the 202 typesetter have in some cases disappeared as well, notably the chess font that SRB has had to reconstruct, but also the Print Out font that was for many purposes quite a reasonable alternative to Courier. The hardware itself, and the specialized software that ran on it, has also gone completely.

On the other hand we have had cause to bless our own pack-rat mentalities. For example, we found the 1983 Linotype font catalogue and we also had a preserved copy of the *troff* source of the vacation memo, with a version of *ditroff* capable of processing it. There are clearly a large number of other digital pack rats, to whom we are grateful, because we have been repeatedly and pleasantly surprised by how much apparently lost information can be found on the web by diligent search and occasional serendipity.

It seems clear that the world needs more archival sites that record useful information. And this is not too difficult in the modern era of cheap computer storage. The recorded information needs to include data, data formats, and programs for processing them. Almost any modern document is a complex amalgam of components that depend on other components, so gathering the complete set that is necessary to recreate it is exceptionally difficult. The authors of this paper have seen this in books, technical papers and programs, and of course in hardware of all sorts. Perhaps this paper will serve as a kind of reminder of the importance of saving *everything*, in one’s best guess about formats that will last.

8. ACKNOWLEDGEMENTS

Profound thanks are due to Chuck Bigelow, who encouraged us to undertake this project and who gave endless tutorial advice to DFB during hours of work on the PO

font. Thanks also to Ken Thompson for the information he supplied about his chess font, and to Andy Walker for pointing us to reference [11].

It will be readily apparent just how much a project of this sort is indebted to John Warnock, and his colleagues at Adobe Systems Inc., for developing PostScript and PDF.

Thanks to the invaluable cooperation of Lucie Cohn and Ed Hummel at Alcatel-Lucent (the parent of Bell Labs today), the rebuilt paper can now be seen at [2].

9. REFERENCES

- [1] B. W. Kernighan, “A Typesetter Independent TROFF,” Computing Science Technical Report No. 97, Bell Laboratories, Murray Hill, New Jersey 07974, March 1982.
- [2] Joe Condon, Brian Kernighan, and Ken Thompson, “Experience with the Mergenthaler Linotron 202 Phototypesetter, or How We Spent Our Summer Vacation,” Computing Science Technical memorandum, January 6, 1980. Available online (rebuilt version) at: <http://www.cs.princeton.edu/~bwk/202>
- [3] Joe Condon, Brian Kernighan, and Ken Thompson, “Experience with the Mergenthaler Linotron 202 Phototypesetter, or How We Spent Our Summer Vacation,” Computing Science Technical memorandum, January 6, 1980. Available online (original page scans) at: <http://www.cs.princeton.edu/~bwk/202>
- [4] D F Brailsford, “In-house production of Examination Papers using *troff*, *eqn* and *tbl*,” in *Proceedings PROTEXT I Workshop, Dublin*, pp. 21–28, Boole Press, October 1984. <http://eprints.nottingham.ac.uk/363/1/protex.pdf>
- [5] David F. Brailsford, “Automated Re-typesetting, Indexing and Content Enhancement for Scanned Marriage Registers,” in *Proceedings of the ACM Symposium on Document Engineering (DocEng09)*, pp. 29–38, ACM Press, 15–18 September 2009.
- [6] Myfonts.com, *WhatTheFont*: font identification software. <http://new.myfonts.com/WhatTheFont>.
- [7] B. W. Kernighan, “PIC — A Language for Typesetting Graphics,” *Software — Practice and Experience*, vol. 12, no. 1, pp. 1–21, January, 1982.
- [8] N. Batchelder and Trevor Darrell, *Pstfig — A Ditroff Preprocessor for PostScript files*, Computer and Information Science Dept., University of Pennsylvania, 1988. Internal Report.
- [9] Martin W. Guy, Various 202 font widths and other *ditroff* programs and resources. <http://medialab.freaknet.org/martin/tape/stuff/ditroff/>
- [10] Ken Thompson, Personal Communication, 8th December 2012.
- [11] David Levy, Kevin O’Connell, and David Watt, *The Sicilian Defence*, Imprint Editions, 1981. <http://www.amazon.co.uk/dp/0907352006>