FOREWORD

The attached paper appeared as part of a Workshop session of tutorial lectures held at the PRO-TEXT I Conference in late 1984. The lecture was an informal presentation telling the story of how the University of Nottingham equipped itself — just prior to the era of PostScript and "desktop publishing" — to typeset all of its examination papers 'in house'. On a personal level, my volunteering to lead this project triggered an interest in computerised typesetting which later developed into a new direction in my computer science research interests. This 'new direction' has now lasted for more than 20 years and has grown to encompass almost all aspects of a subject now called 'document engineering'.

The paper is of some historical interest, if only to note that the equipment we used cost us the staggering sum of £80,000 in 1982. Nowadays comparable quality, and much greater speed, could be obtained from a computer and laser-printer combination costing about one twentieth of that amount.

The key factors in the success of our project, in those very early days, were the willingness of the University not only to buy the necessary typesetters and computers but also to purchase a source code licence for the device-independent version of *troff*. This latter item, at £4000, was not cheap but it has done sterling service and is still in use, some 20 years later, in the School of Computer Science. Above all, the device independent *troff*, (unlike the original version of *troff* which was for the second-generation GSI CAT typesetter only) proved to be readily adaptable to a variety of typesetters and laser-printers. This meant, in turn, that the only other readily available software for device-independent typesetting (Don Knuth's TEX) did not need to be considered. Ruling out the use of TEX was no reflection on its considerable typesetting capabilities but simply a pragmatic reaction to the fact that TEX needed a DEC VAX, rather than a PDP11, and budget constraints absolutely ruled out this sort of expenditure.

COLOPHON

As stated in the 'Acknowledgements' section this paper was originally coded up in *troff*, by Marion Windsor, in early 1984. It was then typeset on bromide using the same Linotype 202 machine described in the paper. The bromide was submitted to the Conference Proceedings as 'camera ready copy'. The source text of the paper was then lost during a transition from VAX-based to SUN-based UNIX systems in the late 1980s (a sobering reminder of the importance of rigorous archiving policies …). This rebuilt form of the paper was obtained by scanning in from the Conference Proceedings and then using Readiris OCR on the resulting TIFF files. The paper was re-typeset using UNIX *troff* to set up the correct typeface (Times). Equations and tables were re-set using the *eqn* and *tbl* pre-processors for *troff*. and the opportunity has been taken to correct a few typographic and factual errors in the original paper.

The time taken to rebuild this paper (over several lunchtimes …) was about 4 hours; a large proportion of that time was spent in ensuring the accuracy of the examination paper front sheet shown in Figure 1.

In-house Preparation of Examination Papers using *troff*, *tbl*, and *eqn*

David F. Brailsford

Computer Science Group
University of Nottingham
NOTTINGHAM NG7 2RD
U.K.

ABSTRACT

Starting in December 1982 the University of Nottingham decided to phototypeset almost all of its examination papers 'in house' using the *troff*, *tbl* and *eqn* programs running under UNIX. This tutorial lecture highlights the features of the three programs with particular reference to their strengths and weaknesses in a production environment. The following issues are particularly addressed:

> **Standards** — all three software packages require the embedding of commands and the invocation of pre-written macros, rather than 'what you see is what you get'. This can help to enforce standards, in the absence of traditional compositor skills.

> **Hardware and Software** — the requirements are analysed for an inexpensive preview facility and a low-level interface to the phototypesetter.

> **Mathematical and Technical papers** — the fine-tuning of eqn to impose a standard house style.

> **Staff skills and training** — systems of this kind do not require the operators to have had previous experience of phototypesetting. Of much greater importance is willingness and flexibility in learning how to use computer systems.

1. Introduction and Background Information

It is a rather strange experience for an academic to be thrown into the role of storyteller and yet, in this tutorial paper, I shall attempt to convey the philosophy of the *troff*, *tbl* and *eqn* software running under UNIX and to relate our experiences in using it, in a production environment, at the University of Nottingham for the in-house phototypesetting of a wide variety of material. Although we have set posters, newsletters, journals, pamphlets and forms using this software, the majority of the work has been in the production of examination papers and I shall concentrate on this area in what follows.

It will soon be evident that setting up a system of this nature in less than a year, and conditioning it for non-expert use, needs coordinated, time-critical, teamwork of an order normally associated only with jugglers and acrobats. From time to time those particular professions seemed strangely appropriate as we struggled with software bugs which insisted on replacing line-end characters by strings of Ws, or with phototypesetters which pumped toner ink all over our feet. Nevertheless the results we have achieved make us feel it was all worth while and a tribute to the rest of the team appears in section 2 and in the 'Acknowledgements' section.

In common with many other groups, our interest in phototypesetting began when we received the UNIX operating system and began to run it on our DEC PDP 11/34 computer. Quite apart from the merits of UNIX as an operating system we were intrigued by the text preparation packages *nroff* (for low resolution terminals and printers) and *troff* (for phototypesetter output on a Graphics Systems Inc. CAT machine).

The *nroff* and *troff* programs [Oss1979a] have much code in common but we shall be concentrating on *troff* from now on. The input format accepted by *troff* is the classic one of having layout commands embedded within the text. These commands usually occur at the start of a line and begin with a . character. Other commands (some of which are merely alternative forms of commands) can occur within a line and are preceded by a \ character. So, for example, the text

```
.ps 10
.vs 12
.ft HR
This should be in Helvetica whereas this switches back to \f(TRTimes.
```

asks for a point size of 10 and a vertical line spacing of 12 points followed by a change into the Helvetica Roman font. Part way through the sentence *troff* recognises \f(TR as a request to change font to Times Roman. The net effect is to produce

This should be in Helvetica whereas this switches back to Times.

Troff is amply endowed with dozens of low-level commands for asserting fine control over factors such as horizontal and vertical spacing, hyphenation and the like. To use the input language in this raw form requires considerable expertise and tenacity, with the danger that the traditional typesetting technique of making fine adjustments by inserting lead spacers is replaced by the even more tedious drudgery of calculating such spacings in ems or in machine units.

Clearly, forms of input are needed which express more abstractly and powerfully what the intended effect is to be. This is accomplished in two ways. Firstly, *troff* has a macro facility which enables frequently used sequences of commands to be encapsulated in a single command name. Files of standard macro definitions can be built up and two useful sets, supplied with UNIX, are 'ms' (for general typing of manuscripts) and 'man' (for typesetting the UNIX manual pages). Macro commands, like the raw *troff* commands, are preceded by a . but the usual convention for macros is that upper case names are used. As with macro systems in other areas of computer software, the power of a macro approach becomes especially evident when they can be supplied with parameters (or arguments) which control the precise effect obtained. Thus if the .IP (indented paragraph) command of the 'ms' macro package is invoked by typing

```
.IP hello 7
```

then the correct spacing motions will occur to set an indented paragraph with hanging tag 'hello' and with an indentation of 7 ens.

The second method of making *troff* accessible to mere apprentices is to provide pre-processors which allow the user to specify, in a high-level fashion, what is required and which will translate these wishes into *troff* commands to effect them. *Tbl* and *eqn* follow precisely this philosophy. The former [Les1979a] allows templates for tables to be set up and for tabulation commands to be given. It recognises .TS ('Table Start') and .TE ('Table End') as being delimiters in the input text between which all typesetting instructions will be in *tbl* format. Similarly, *eqn* [Ker1975a] expects all commands between .EQ and .EN, for displayed equations and between $...$ for in-line equations in the 'eqn language'. As a simple example of what can be done the input

```
.EQ
{gamma + delta sub 2} over sqrt {a sup 2 + b sup 2}
.EN
```

produces

$$\frac{\gamma + \delta_2}{\sqrt{a^2 + b^2}}$$

with all calculations of the heights and widths of fraction bar and vinculum being performed automatically. It goes without saying that preprocessors of this sort can outdo even the most demented human in terms of the sheer quantity, fluency and dazzling obscurity of the raw *troff* output they generate.

2. The Examinations Project

Early in 1982 we were asked by the Examinations Unit at our University to assess whether a *troff* based system would be suitable for in-house installation, with the primary aim of typesetting all examination papers over the crucial period from January to May every year. At other times of the year a typesetting service would be provided for other University material.

At that time, the annual cost of sending examination papers to outside organisations for typesetting was in the region of £18,000 p.a. The situation was particularly aggravated by the requirements of the Mathematics degree examinations where, rather than setting a small number of papers at the Final Examination, an option scheme had been adopted which produced a separate two-hour paper for each of about 50 possible options. It will be no surprise to learn that the typesetting and correction of these papers alone would occupy an experienced compositor for fully 6 weeks.

Somewhat nervously our University set aside £35,000 to purchase a pilot system consisting of DEC LSI 11/23 computer, Linotype Omnitech 2000 phototypesetter and an AT&T Phototypesetter License. An implementation team was set up with the following duties:

D.F. Brailsford	(project manager; computer and phototypesetter interfacing; initial version of 'mex' macros)
D.R Woodall	(installation and software management of *troff*, *tbl* and *eqn*)
G.F.Paechter	(typographical specifications; technical editor; 'mex' macro enhancements)
W.J. Armitage	(installation and maintenance of UNIX software; assistance
J.P. Onions	with interfacing problems)
S.J. Marchant	design of custom hardware, with Z80 controllers, to interface to parallel ports on Omnitech and Linotype 202 typesetters

3. Software development

3.1. Modifications to *troff*

At this early stage we were still using the original *troff* program, as supplied with Version 7 of the UNIX system, whose output was for GSI CAT machine. A considerable amount of work went into rewriting *troff* to output codes which would satisfy the T-file format required for driving the Omnitech 2000. A few extra features were also added, in the form of new raw commands, notably a mechanism for establishing modes such as bold, italic or negative area. Thus, for example

```
.md B
This is bold
.md-B
```

would change into bold-face but would retain the underlying type style (Times, Helvetica etc.). This obviates the need to stipulate the explicit fonts being used and by re-initialising the type style in use at the head of the document enables the same material to be produced in a variety of typefaces without altering the basic text. A problem which this approach raises is what to do when a given style is not available for the font in use. The solution adopted has been to produce

simulated italic via the 'slope forward' facility available on many modern setters. Simulated emboldening is achieved by overstriking the character 3 times with a tiny horizontal displacement on each occasion. The effect this produces is just about acceptable for sans-serif fonts but is of course, no substitute for possessing the proper bold font. The facility has been used, in the main, for emboldening certain mathematical characters that were not available in bold form on any of our fonts and which needed emboldening when used to denote vectors.

3.2. The mex Macros

An example of a front page of one of our examination papers is shown in Figure l. It will be immediately apparent that certain standard features such as headings, underlines, indentations of question numbers, the phrase *Turn Over* on all pages except the last, and so on, can be set up as the standard 'house style'. A macro package called 'mex' (**M**acros for **Ex**ams)was written and has evolved steadily over the past two years. For example the first few lines of Figure 1 were obtained by typing

```
.EX 1144
.EQ
delim
global cent
.EN
.UN
.FU S
.ce
Physics
.sp
.ET 10 1983
.ce
PHYSICS HIII (iii)
.sp
.TI 2
.CM
.CR
.PH
```

3.3. Modifications to *eqn*

Unlike *troff*, the *eqn* preprocessor has only a rudimentary macro capability and the imposition of a house style becomes somewhat trickier. Certain spacing defaults and actions are inherent in the implementation of *eqn* keywords, but it is true to say that there is considerable scope for adjusting the spacing of elements within the equation by use of the ˜ and ˆ characters to denote full and half-spaces respectively. In principle this gives plenty of control over layout but means that each equation might possess many such marks to ensure a pleasant appearance. Uncontrolled use of these spacing characters results in very diffuse and displeasing effects.

With the production of examination papers in mind our resident typographical experts decided to tune the *eqn* program with a view to imposing a compact equation style reminiscent of that found in [Cha1954a]. Many default spacings associated with the centring of equations, space around = signs and so on were adjusted so as to minimise the number of explicit ˆ and ˜ marks needed when typing in the paper.

University of Nottingham

FACULTY OF SCIENCE

Physics

PART IIB EXAMINATION, 1983

PHYSICS HIII (iii)

Time allowed TWO hours

Candidates must NOT start writing their answers until told to do so

Candidates may use a self-contained silent electronic calculator in this examination, provided that the make and type number are noted on the front of the script. Programmable calculators must not be programmed prior to the examination, and no program in any form may be taken into the examination room.

An indication is given of the approximate weighting of each part of a question by means of a bold figure enclosed by curly brackets, e.g. {**2**}, immediately following that part

Marks will be given for the following where appropriate: understanding the problem; use of relevant equations; obtaining an answer which is plausible; making realistic estimates for those quantities that are not given; for clear well labelled diagrams and for practical solutions to the experimental problems. ALL questions may be attempted. Total marks should be regarded as out of 80. The following data may (or may not) prove useful.

electronic charge	e	=	$1.602\,1918 \times 10^{-19}$ C
electronic rest mass	m_e	=	$0.9109547 \times 10^{-30}$ kg
proton's rest mass	m_p	=	1.6726×10^{-27} kg
neutron rest mass	M_n	=	1.6748×10^{-27} kg
Planck's constant	h	=	$6.626\,1933 \times 10^{-34}$ J s
speed of light	c	=	$2.997\,924\,580 \times 10^{8}$ m s^{-1}
acceleration due to gravity	g	=	9.8 m s^{-2}
Avogadro's number	N_A	=	$6.022\,098 \times 10^{26}$ kmol^{-1}
gas constant	R	=	8.3145 J K^{-1}mol^{-1}
Rydberg constant	R_y	=	13.6058 eV
Boltzmann's constant	k_B	=	$1.380\,50 \times 10^{-23}$ J K^{-1}
Bohr magneton	μ_B	= $\dfrac{e\hbar}{2m_e}$	= $9.274\,10 \times 10^{-24}$ J T^{-1}
Stefan's constant	σ	=	5.6703×10^{-18} W m^{-2} K^{-4}
gravitational constant	G	=	6.673×10^{-11} N m^2 kg^{-2}
fine-structure constant	α	= $\dfrac{e^2}{4\pi\varepsilon_0\hbar c}$	= $(137.035\,96)^{-1}$
permittivity of free space	ε_0	=	8.85419×10^{-12} F m^{-1}
permeability of free space	μ_0	=	$4\pi \times 10^{-7}$ H m^{-1}
radius of the earth	R_E	=	6.378×10^{6} m
molecular weight of air		=	28.8 g mol^{-1}
γ ($=c_p/c_v$) for air		=	1.40

Figure 1: Front page of a sample examination paper

This approach marks out a rather different philosophy than the one that is implicit in the original program. Perhaps the only way to reconcile the two is for *eqn* to be re-written so as to externalise, as string definitions or as tables, all the parameters which might potentially need to be adjusted to accommodate a given house-style. But it has to be admitted that attempts to make eqn more independent of the ultimate output device are not as easy as would appear at first sight. Ideally, of course, *eqn* simply feeds its output into *troff*, which ultimately drives the phototypesetter. In practice many of the details of the particular mathematical symbols available and their sizes will inevitably affect *eqn*'s strategy. Good example of this can be seen with integral signs, ∫, and square brackets [. It is clear from the source text of *eqn* that, on the original GSI CAT typesetter at Bell Laboratories, these are composite symbols made up from a 'kit of parts' comprising vertical and horizontal bars, arc segments and so on. On our Linotype fonts however, we possess each of these symbols explicitly at 3 different sizes. Heated arguments took place, of almost theological intensity, as to where the overlap should occur, in point size terms, between magnifying a small version of ∫ and diminishing a larger one — and whether the stem weights were then acceptable, and so on.

4. Hardware

The pilot project from April 1982–November 1983 was conducted on a LSI 11/23 running UNIX and driving an Omnitech 2000 phototypesetter. This system was successful in producing all the technical examination papers for 1983. Its performance was limited by the slowness of the LSI II even when driving only two terminals but even more severely by the slowness and unreliability of the Omnitech. This particular typesetter made a brave and far-sighted attempt to dispense with bromide and to use a laser driven raster scan system to set text onto electrically charged paper. The overall technology is akin to photocopying in many respects but the problems of backlash on the drum drive and leaks in the plumbing for the toner ink nearly drove us to distraction. Fortunately, this machine has now been withdrawn from the market to be replaced by the much more reliable Linotype 101.

For our part we replaced the pilot system in November 1983 with a PDP 11/44 computer driving a Linotype 202 phototypesetter. Although based on older technology, the 202 positively gallops through the work and the 11/44 copes fairly cheerfully with driving up to six terminals and the 202 typesetter. The Omnitech 2000, now put out to pasture in the Main Library, is also driven from the 11/44 via a remote serial link.

5. Interfacing

5.1. Hardware

There have been few major problems here. For reasons of price and performance we have purchased parallel interfaces on both our typesetters but we drive them down a conventional multiplexer serial line. The necessary serial-to-parallel conversion, buffering and parity checking is carried out by a Z80 single- board computer. The software for this interface was written using the Whitesmiths C cross-compiler, hosted on the PDP11 and LSI11, and producing code for the Z80. This code was, in turn, burned onto an EPROM mounted on the Z80 single-board computer.

5.2. Software

After spending almost two years modifying *troff* you can imagine the software implementers' joy at discovering that a more modern, and typesetter-independent, version of *troff* had existed all along. This version, [Ker1982a] now known as DITROFF (**D**evice **I**ndependent *troff*), produces a low-level device-independent code for driving a variety of output devices. We have christened this code DIC (**D**itroff **I**ntermediate **C**ode) and its presence makes the interfacing of

new output devices much easier than before. A 'back end' program has to be written to convert the DIC into the particular codes required by each target phototypesetter, but with example code available from other driver programs this enables a working driver to be fabricated in about a day, and a sophisticated one within a week.

The over-riding requirement for any software running in the typesetter is that it must be at a similar low level of abstraction as DIC itself and should allow the setter to be driven in a slave mode. That is to say, it should support absolute and relative spacing commands at the full resolution of the machine. Suitable primitives for point size change, font change, reverse area etc. should also be supplied. Any higher-level software which, for example, insists on re-justifying the text which *troff* has already justified, or which limits the accuracy with which superscripts and subscripts can be placed, should be politely but firmly refused.

Fortunately the T-file language on the Omnitech and the Binary Byte language on the 202 have been at about the right level and have proved very suitable for our needs.

6. Preview facilities

One lingering problem that faces us is the provision of preview facilities, at modest cost, for our operators, so that the final phototypeset effect can be judged and proofread before committing it to bromide. We have had some limited success in this area using DEC VT102 terminals equipped with Digital Engineering Retrographics units but reliability problems and the slowness of the processor on the Retrographics board have held us up. More sophisticated preview facilities are available under UNIX System V on the BLIT (now Teletype 5620) terminal but these are still rather expensive at the time of writing.

7. Operator experiences

The handful of people who control the keyboarding of the examination papers have all had a good educational background but do not possess any particular mathematical skills. We were fortunate, in the early days, to be able to recruit a person with some experience of computers, typing and the publishing trade. But with only a few exceptions even the later, inexperienced, operators have learned very quickly how to transform an initial handwritten or typewritten version of an examination paper into a mixture of text, *troff* commands and *eqn* commands.

The originals for the paper are marked up by our technical editor in *troff* and *eqn* notation, but only to give occasional guidance where a rather abstruse combination of commands is needed to achieve the desired effect. It can be noted at this point that a potent advantage of the *troff*/*eqn* embedded commands approach is that standards can be imposed and maintained. If the paragraph indentation caused by a call of the `.PP` macro is inappropriate or if an *eqn* command such as `$a sup 2$` results in an improperly placed superscript then we know it is not a mistake on the operator's part but instead an indication of the need for an adjustment in the details of the `.PP` and `sup` implementations.

8. User reactions

Later stages of proof reading are carried out by the technical editor and ultimately by the lecturer who set the paper. There is widespread appreciation of the fast turnround of proofs especially since the commissioning of the 202 and there is broad agreement that out standards now meet or exceed those which we obtained when our papers were sent outside the University for setting.

There was some initial confusion among proof-readers, during the early stages of tuning the software, when an over-zealous tweaking of some software parameter might cause a torrent of errors to appear in a paper that had previously been all right. It proved quite hard to convince members

of academic staff that the perceived errors were of a similar nature and that they would all disappear, simultaneously, after a little more surgery on the 'mex' macros or on *eqn*.

By contrast, of course, a more interactive scheme with 'what you see is what you get', without macro assistance, is closer to the traditional way of setting type, but demands intense concentration to ensure that the house-style is maintained.

9. Conclusions

By any standards our experiment has to be judged a success. The savings made by our University in typesetting examination papers and other material in-house will be sufficient to nullify the £80,000 cost of the PDP 11/44 computer and Linotype 202 phototypesetter after only two years. We see further potential benefits in future from the UNIX/DITROFF combination. For example, the UNIX environment would enable a database of student records to be kept on disc and, at appropriate times, items such as degree certificates, registration cards etc. could be produced on a range of output devices such as phototypesetters and laser printers.

So far there has been a lot of interest in the use of the UNIX phototypesetting software. However, the market for such technically orientated setting is rather limited and so there are few if any turnkey systems available. With this in mind it is clear that much of our success can be attributed to the fact that although we lacked some of the traditional composing and typesetting skills we did have to hand a fair body of expertise in mathematical typography, mathematics and computer science.

A system such as ours, as it stands, would perhaps be unacceptable to 'the trade' where a more interactive, 'wysiwyg', approach might be favoured. Nevertheless we feel sufficiently convinced of the merits of *describing* what you want, in *eqn* notation rather than the more difficult task of maintaining standards and style when repetitively *doing* what you want, that we are happy to maintain the approach of tagged input to *troff* and *eqn*, provided that we can have an output preview capability of some sort before committing ourselves to the creation of film or bromide.

10. Acknowledgements

I should like to thank the Registrar and Deputy Registrar of the University of Nottingham for suppressing any misgivings they may have had and for supporting us in this project. The Examinations Unit supervisor, Betty Hickling, and all her staff, have remained stoically cheerful through many software changes, floods of ink and the occasional bolt of lightning to the PDP 11/44. Brian Kernighan provided much helpful advice on his typesetter-independent version of *troff* together with a fascinating technical report of work at Bell Laboratories in 'reverse engineering' the Linotype 202 so that it could be driven, as a slave device, at a coding level even lower than Binary Byte.

Thanks also to Marion Windsor for typing this paper and for her hard work in the early days of commissioning the Examinations system. I do not dare to add up the man-hours contributed by those named in Section 2, but it goes without saying that the successful outcome of this project is a tribute to many late-night sessions labouring over source code, macro modifications and hardware interface testing.

References

Oss1979a. J.F. Ossanna, "NROFF/TROFF User's Manual", *UNIX Programmer's Manual* **2**, Section 22 (January 1979).

Les1979a. M.E. Lesk, "Tbl — A Program to Format Tables", *UNIX Programmer's Manual* **2**, Section 10 (January 1979).

Ker1975a. B.W. Kernighan and L.L. Cherry, "A System for Typesetting Mathematics", *Comm. ACM* **18**(3), pp.151–157 (1975).

Cha1954a. T.W. Chaundy, P.R. Barrett, and C. Batey, *The Printing of Mathematics*, Oxford University Press, Oxford (1954).

Ker1982a. B.W. Kernighan, "A Typesetter-independent TROFF", Comp. Sci. Tech. Rep., Bell Laboratories, Murray Hill, NJ (March 1982).