

Martin Richards

How to write a dissertation in L^AT_EX

Computer Science Tripos – Part II

St John's College

October 11, 2018

Proforma

Name:	Martin Richards
College:	St John's College
Project Title:	How to write a dissertation in L^AT_EX
Examination:	Computer Science Tripos – Part II, July 2001
Word Count:	1587¹ (well less than the 12000 limit)
Project Originator:	Dr M. Richards
Supervisor:	Dr Markus Kuhn

Original Aims of the Project

To write a demonstration dissertation² using L^AT_EX to save student's time when writing their own dissertations. The dissertation should illustrate how to use the more common L^AT_EX constructs. It should include pictures and diagrams to show how these can be incorporated into the dissertation. It should contain the entire L^AT_EX source of the dissertation and the makefile. It should explain how to construct an MSDOS disk of the dissertation in Postscript format that can be used by the book shop for printing, and, finally, it should have the prescribed layout and format of a diploma dissertation.

Work Completed

All that has been completed appears in this dissertation.

Special Difficulties

Learning how to incorporate encapsulated postscript into a L^AT_EX document on both Ubuntu Linux and OS X.

¹This word count was computed by `detex diss.tex | tr -cd '0-9A-Za-z \n' | wc -w`

²A normal footnote without the complication of being in a table.

Declaration

I, [Name] of [College], being a candidate for Part II of the Computer Science Tripos [or the Diploma in Computer Science], hereby declare that this dissertation and the work described in it are my own work, unaided except as may be specified below, and that the dissertation does not contain material that has already been used to any substantial extent for a comparable purpose.

Signed [signature]

Date [date]

Contents

1	Introduction	9
1.1	Overview of the files	9
1.2	Building the document	9
1.2.1	The makefile	9
1.3	Counting words	10
2	Preparation	11
3	Implementation	13
3.1	Verbatim text	13
3.2	Tables	14
3.3	Simple diagrams	14
3.4	Adding more complicated graphics	14
4	Evaluation	17
4.1	Printing and binding	17
4.2	Further information	17
5	Conclusion	19
	Bibliography	19
A	Latex source	23
A.1	diss.tex	23
A.2	proposal.tex	29
B	Makefile	33
B.1	makefile	33
B.2	refs.bib	33
C	Project Proposal	35

List of Figures

3.1	A picture composed of boxes and vectors.	14
3.2	A diagram composed of circles, lines and boxes.	15
3.3	Example figure using encapsulated postscript	15
3.4	Example figure where a picture can be pasted in	16
3.5	Example diagram drawn using <code>xfig</code>	16

Acknowledgements

This document owes much to an earlier version written by Simon Moore [2]. His help, encouragement and advice was greatly appreciated.

Chapter 1

Introduction

1.1 Overview of the files

This document consists of the following files:

- `makefile` — The makefile for the dissertation and Project Proposal
- `diss.tex` — The dissertation
- `proposal.tex` — The project proposal
- `figs` — A directory containing diagrams and pictures
- `refs.bib` — The bibliography database

1.2 Building the document

This document was produced using $\text{\LaTeX} 2_{\epsilon}$ which is based upon \LaTeX [1]. To build the document you first need to generate `diss.aux` which, amongst other things, contains the references used. This is done by executing the command:

```
pdflatex diss
```

Then the bibliography can be generated from `refs.bib` using:

```
bibtex diss
```

Finally, to ensure all the page numbering is correct run `pdflatex` on `diss.tex` until the `.aux` files do not change. This usually takes 2 more runs.

1.2.1 The makefile

To simplify the calls to `pdflatex` and `bibtex`, a makefile has been provided, see Appendix B.1. It provides the following facilities:

```
make
```

Display help information.

make proposal.pdf

Format the proposal document as a PDF.

make view-proposal

Run **make proposal.pdf** and then display it with a Linux PDF viewer (preferably “okular”, if that is not available fall back to “evince”).

make diss.pdf

Format the dissertation document as a PDF.

make count

Display an estimate of the word count.

make all

Construct **proposal.pdf** and **diss.pdf**.

make pub

Make **diss.pdf** and place it in my **public.html** directory.

make clean

Delete all intermediate files except the source files and the resulting PDFs. All these deleted files can be reconstructed by typing **make all**.

1.3 Counting words

An approximate word count of the body of the dissertation may be obtained using:

```
wc diss.tex
```

Alternatively, try something like:

```
detex diss.tex | tr -cd '0-9A-Z a-z\n' | wc -w
```

Chapter 2

Preparation

This chapter is empty!

Chapter 3

Implementation

3.1 Verbatim text

Verbatim text can be included using `\begin{verbatim}` and `\end{verbatim}`. I normally use a slightly smaller font and often squeeze the lines a little closer together, as in:

```
GET "libhdr"

GLOBAL { count:200; all  }

LET try(ld, row, rd) BE TEST row=all
      THEN count := count + 1
      ELSE { LET poss = all & ~(ld | row | rd)
            UNTIL poss=0 DO
              { LET p = poss & -poss
                poss := poss - p
                try(ld+p << 1, row+p, rd+p >> 1)
              }
            }

LET start() = VALOF
{ all := 1
  FOR i = 1 TO 12 DO
  { count := 0
    try(0, 0, 0)
    writef("Number of solutions to %i2-queens is %i5*n", i, count)
    all := 2*all + 1
  }
  RESULTIS 0
}
```

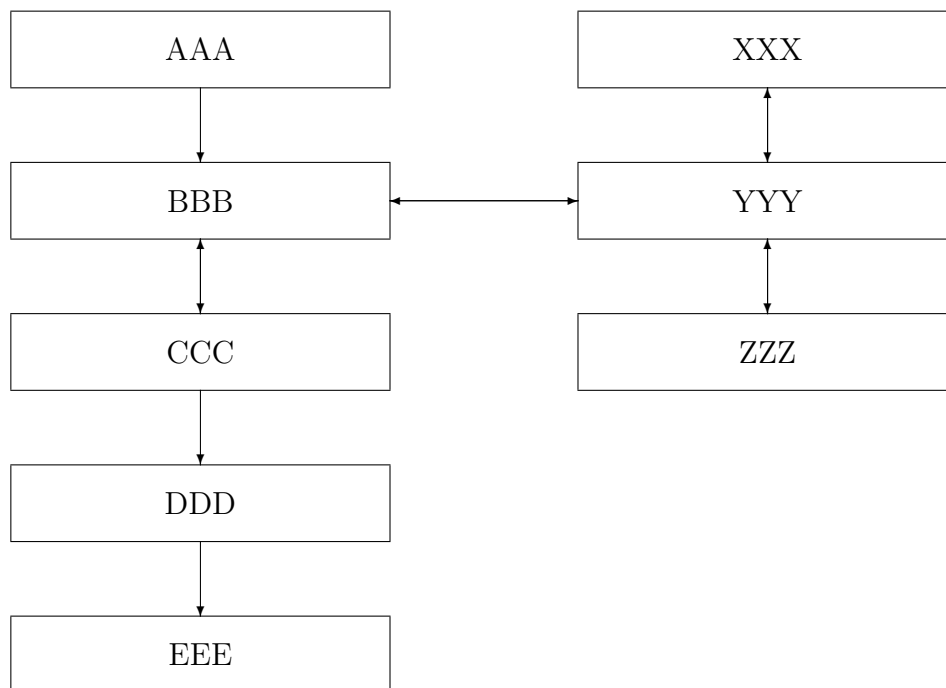


Figure 3.1: A picture composed of boxes and vectors.

3.2 Tables

Here is a simple example¹ of a table.

Left Justified	Centred	Right Justified
First	A	XXX
Second	AA	XX
Last	AAA	X

There is another example table in the proforma.

3.3 Simple diagrams

Simple diagrams can be written directly in \LaTeX . For example, see figure 3.1 on page 14 and see figure 3.2 on page 15.

3.4 Adding more complicated graphics

The use of \LaTeX format can be tedious and it is often better to use encapsulated postscript (EPS) or PDF to represent complicated graphics. Figure 3.3 and 3.5 on page 16 are

¹A footnote



Figure 3.2: A diagram composed of circles, lines and boxes.

examples. The second figure was drawn using `xfig` and exported in `.eps` format. This is my recommended way of drawing all diagrams.



Figure 3.3: Example figure using encapsulated postscript

Figure 3.4: Example figure where a picture can be pasted in

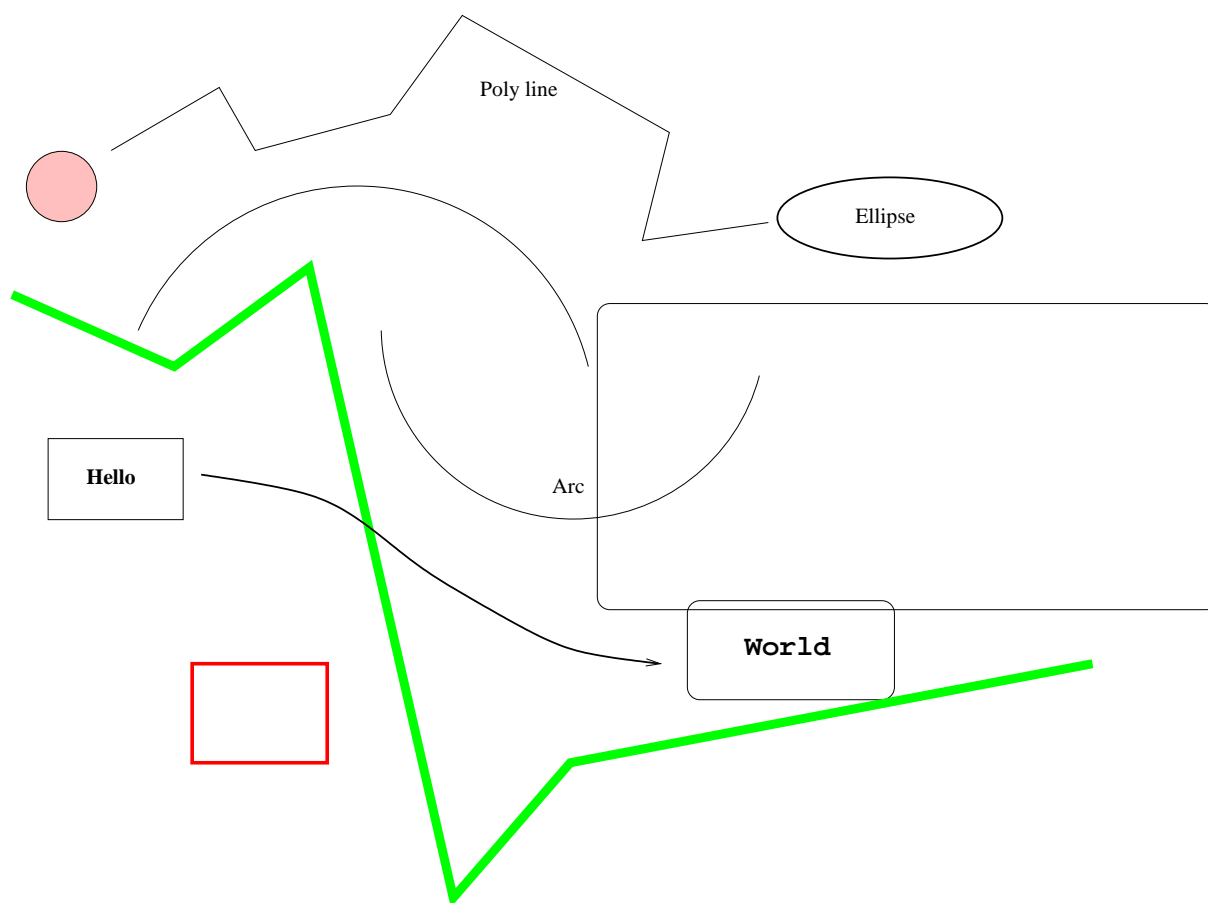


Figure 3.5: Example diagram drawn using `xfig`

Chapter 4

Evaluation

4.1 Printing and binding

Use a “duplex” laser printer that can print on both sides to print two copies of your dissertation. Then bind them, for example using the comb binder in the Computer Laboratory Library.

4.2 Further information

See the Unix Tools notes at

<http://www.cl.cam.ac.uk/teaching/current-1/UnixTools/materials.html>

Chapter 5

Conclusion

I hope that this rough guide to writing a dissertation in L^AT_EX has been helpful and saved you time.

Bibliography

- [1] L. Lamport. *LaTeX — a document preparation system — user's guide and reference manual*. Addison-Wesley, 1986.
- [2] S.W. Moore. How to prepare a dissertation in latex, 1995.

Appendix A

Latex source

A.1 diss.tex

```
% Template for a Computer Science Tripos Part II project dissertation
\documentclass[12pt,a4paper,twoside,openright]{report}
\usepackage[pdftborder={0 0 0}]{hyperref} % turns references into hyperlinks
\usepackage[margin=25mm]{geometry} % adjusts page layout
\usepackage{graphicx} % allows inclusion of PDF, PNG and JPG images
\usepackage{verbatim}
\usepackage{docmute} % only needed to allow inclusion of proposal.tex

\raggedbottom % try to avoid widows and orphans
\sloppy
\clubpenalty1000%
\widowpenalty1000%

\renewcommand{\baselinestretch}{1.1} % adjust line spacing to make
% more readable

\begin{document}

\bibliographystyle{plain}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Title

\pagestyle{empty}

\rightline{\LARGE \textbf{Martin Richards}}

\vspace*{60mm}
\begin{center}
\Huge
\textbf{How to write a dissertation in \LaTeX} \\[5mm]
Computer Science Tripos -- Part II \\[5mm]
St John's College \\[5mm]
\today % today's date
\end{center}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Proforma, table of contents and list of figures

\pagestyle{plain}

\chapter*{Proforma}
```

```

{\large
\begin{tabular}{ll}
Name: & & \bf Martin Richards & \\
College: & & \bf St John's College & \\
Project Title: & & \bf How to write a dissertation in \LaTeX & \\
Examination: & & \bf Computer Science Tripos -- Part II, July 2001 & \\
Word Count: & & \bf 1587\footnotemark[1] & \\
& & (well less than the 12000 limit) & \\
Project Originator: & & Dr M.~Richards & \\
Supervisor: & & Dr Markus Kuhn & \\
\end{tabular}
}
\footnotetext[1]{This word count was computed
by \texttt{detex diss.tex | tr -cd '0-9A-Za-z $\t\backslash$ | wc -w}
}
\stepcounter{footnote}

\section*{Original Aims of the Project}

To write a demonstration dissertation\footnote{A normal footnote without the
complication of being in a table.} using \LaTeX\ to save
student's time when writing their own dissertations. The dissertation
should illustrate how to use the more common \LaTeX\ constructs. It
should include pictures and diagrams to show how these can be
incorporated into the dissertation. It should contain the entire
\LaTeX\ source of the dissertation and the makefile. It should
explain how to construct an MSDOS disk of the dissertation in
Postscript format that can be used by the book shop for printing, and,
finally, it should have the prescribed layout and format of a diploma
dissertation.

\section*{Work Completed}

All that has been completed appears in this dissertation.

\section*{Special Difficulties}

Learning how to incorporate encapsulated postscript into a \LaTeX\
document on both Ubuntu Linux and OS X.

\newpage
\section*{Declaration}

I, [Name] of [College], being a candidate for Part II of the Computer
Science Tripos [or the Diploma in Computer Science], hereby declare
that this dissertation and the work described in it are my own work,
unaided except as may be specified below, and that the dissertation
does not contain material that has already been used to any substantial
extent for a comparable purpose.

\bigskip
\leftline{Signed [signature]}

\medskip
\leftline{Date [date]}

\tableofcontents

\listoffigures

\newpage
\section*{Acknowledgements}

This document owes much to an earlier version written by Simon Moore
\cite{Moore95}. His help, encouragement and advice was greatly

```


appreciated.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% now for the chapters
```

```
\pagestyle{headings}
```

```
\chapter{Introduction}
```

```
\section{Overview of the files}
```

This document consists of the following files:

```
\begin{itemize}
\item \texttt{makefile} --- The makefile for the dissertation and
                        Project Proposal
\item \texttt{diss.tex} --- The dissertation
\item \texttt{proposal.tex} --- The project proposal
\item \texttt{figs} -- A directory containing diagrams and pictures
\item \texttt{refs.bib} --- The bibliography database
\end{itemize}
```

```
\section{Building the document}
```

This document was produced using `\LaTeXe` which is based upon `\LaTeX\cite{Lamport86}`. To build the document you first need to generate `\texttt{diss.aux}` which, amongst other things, contains the references used. This is done by executing the command:

```
\texttt{pdflatex diss}
```

```
\noindent
```

Then the bibliography can be generated from `\texttt{refs.bib}` using:

```
\texttt{bibtex diss}
```

```
\noindent
```

Finally, to ensure all the page numbering is correct run `\texttt{pdflatex}` on `\texttt{diss.tex}` until the `\texttt{.aux}` files do not change. This usually takes 2 more runs.

```
\subsection{The makefile}
```

To simplify the calls to `\texttt{pdflatex}` and `\texttt{bibtex}`, a makefile has been provided, see Appendix~\ref{makefile}. It provides the following facilities:

```
\begin{description}
```

```
\item\texttt{make} \\\
```

Display help information.

```
\item\texttt{make proposal.pdf} \\\
```

Format the proposal document as a PDF.

```
\item\texttt{make view-proposal} \\\
```

Run `\texttt{make proposal.pdf}` and then display it with a Linux PDF viewer (preferably ‘`okular`’, if that is not available fall back to ‘`evince`’).

```
\item\texttt{make diss.pdf} \\\
```

Format the dissertation document as a PDF.

```
\item\texttt{make count} \\\
```

Display an estimate of the word count.

```
\item\texttt{make all} \\\
```

Construct `\texttt{proposal.pdf}` and `\texttt{diss.pdf}`.

```

\item\texttt{make pub} \ Make \texttt{diss.pdf}
and place it in my \texttt{public\_html} directory.

\item\texttt{make clean} \ Delete all intermediate files except the
source files and the resulting PDFs. All these deleted files can
be reconstructed by typing \texttt{make all}.

\end{description}

\section{Counting words}

An approximate word count of the body of the dissertation may be
obtained using:

\texttt{wc diss.tex}

\noindent
Alternatively, try something like:

\verb|detex diss.tex | tr -cd '0-9A-Z a-z\n' | wc -w/

\chapter{Preparation}

This chapter is empty!

\chapter{Implementation}

\section{Verbatim text}

Verbatim text can be included using \verb|\begin{verbatim}| and
\verb|\end{verbatim}|. I normally use a slightly smaller font and
often squeeze the lines a little closer together, as in:

{\renewcommand{\baselinestretch}{0.8}\small
\begin{verbatim}
GET "libhdr"

GLOBAL { count:200; all  }

LET try(ld, row, rd) BE TEST row=all
      THEN count := count + 1
      ELSE { LET poss = all & ~(ld | row | rd)
            UNTIL poss=0 DO
              { LET p = poss & -poss
                poss := poss - p
                try(ld+p << 1, row+p, rd+p >> 1)
              }
            }

LET start() = VALOF
{ all := 1
  FOR i = 1 TO 12 DO
    { count := 0
      try(0, 0, 0)
      writef("Number of solutions to %i2-queens is %i5*n", i, count)
      all := 2*all + 1
    }
  RESULTIS 0
}
\end{verbatim}
}

\section{Tables}

```

```

\begin{samepage}
Here is a simple example\footnote{A footnote} of a table.

\begin{center}
\begin{tabular}{l|c|r}
Left      & Centred & Right \\
Justified &         & Justified \\
\hline
First     & A       & XXX \\
Second    & AA      & XX \\
Last      & AAA     & X
\end{tabular}
\end{center}

\noindent
There is another example table in the proforma.
\end{samepage}

\section{Simple diagrams}

Simple diagrams can be written directly in \LaTeX. For example, see
figure~\ref{latexpic1} on page~\pageref{latexpic1} and see
figure~\ref{latexpic2} on page~\pageref{latexpic2}.

\begin{figure}
\setlength{\unitlength}{1mm}
\begin{center}
\begin{picture}(125,100)
\put(0,80){\framebox(50,10){AAA}}
\put(0,60){\framebox(50,10){BBB}}
\put(0,40){\framebox(50,10){CCC}}
\put(0,20){\framebox(50,10){DDD}}
\put(0,00){\framebox(50,10){EEE}}

\put(75,80){\framebox(50,10){XXX}}
\put(75,60){\framebox(50,10){YYY}}
\put(75,40){\framebox(50,10){ZZZ}}

\put(25,80){\vector(0,-1){10}}
\put(25,60){\vector(0,-1){10}}
\put(25,50){\vector(0,1){10}}
\put(25,40){\vector(0,-1){10}}
\put(25,20){\vector(0,-1){10}}

\put(100,80){\vector(0,-1){10}}
\put(100,70){\vector(0,1){10}}
\put(100,60){\vector(0,-1){10}}
\put(100,50){\vector(0,1){10}}

\put(50,65){\vector(1,0){25}}
\put(75,65){\vector(-1,0){25}}
\end{picture}
\end{center}
\caption{A picture composed of boxes and vectors.}
\label{latexpic1}
\end{figure}

\begin{figure}
\setlength{\unitlength}{1mm}
\begin{center}

\begin{picture}(100,70)
\put(47,65){\circle{10}}
\put(45,64){abc}

\put(37,45){\circle{10}}
\put(37,51){\line(1,1){7}}

```

```

\put(35,44){def}

\put(57,25){\circle{10}}
\put(57,31){\line(-1,3){9}}
\put(57,31){\line(-3,2){15}}
\put(55,24){ghi}

\put(32,0){\framebox(10,10){A}}
\put(52,0){\framebox(10,10){B}}
\put(37,12){\line(0,1){26}}
\put(37,12){\line(2,1){15}}
\put(57,12){\line(0,2){6}}
\end{picture}

\end{center}
\caption{A diagram composed of circles, lines and boxes.}
\label{latexpic2}
\end{figure}

\section{Adding more complicated graphics}

The use of \LaTeX\ format can be tedious and it is often better to use
encapsulated postscript (EPS) or PDF to represent complicated graphics.
Figure~\ref{epsfig} and~\ref{xfig} on page \pageref{xfig} are
examples. The second figure was drawn using \texttt{xfig} and exported in
\{\tt.eps\} format. This is my recommended way of drawing all diagrams.

\begin{figure}[tbh]
\centerline{\includegraphics{figs/cuarms.pdf}}
\caption{Example figure using encapsulated postscript}
\label{epsfig}
\end{figure}

\begin{figure}[tbh]
\vspace{4in}
\caption{Example figure where a picture can be pasted in}
\label{pastedfig}
\end{figure}

\begin{figure}[tbh]
\centerline{\includegraphics{figs/diagram.pdf}}
\caption{Example diagram drawn using \texttt{xfig}}
\label{xfig}
\end{figure}

\chapter{Evaluation}

\section{Printing and binding}

Use a ‘‘duplex’’ laser printer that can print on both sides to print
two copies of your dissertation. Then bind them, for example using the
comb binder in the Computer Laboratory Library.

\section{Further information}

See the Unix Tools notes at

\url{http://www.cl.cam.ac.uk/teaching/current-1/UnixTools/materials.html}

\chapter{Conclusion}

```

I hope that this rough guide to writing a dissertation is \LaTeX\ has been helpful and saved you time.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% the bibliography
\addcontentsline{toc}{chapter}{Bibliography}
\bibliography{refs}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% the appendices
\appendix

\chapter{Latex source}

\section{diss.tex}
{\scriptsize\verbatiminput{diss.tex}}

\section{proposal.tex}
{\scriptsize\verbatiminput{proposal.tex}}

\chapter{Makefile}

\section{makefile}\label{makefile}
{\scriptsize\verbatiminput{makefile.txt}}

\section{refs.bib}
{\scriptsize\verbatiminput{refs.bib}}

\chapter{Project Proposal}

\input{proposal}

\end{document}

```

A.2 proposal.tex

```

% Note: this file can be compiled on its own, but is also included by
% diss.tex (using the docmute.sty package to ignore the preamble)
\documentclass[12pt,a4paper,twoside]{article}
\usepackage[pdftborder={0 0 0}]{hyperref}
\usepackage[margin=25mm]{geometry}
\usepackage{graphicx}
\usepackage{parskip}

\begin{document}

\begin{center}
\Large
Computer Science Tripos -- Part II -- Project Proposal\\[4mm]
\LARGE
A strongly consistent index for email using git and MirageOS.

\large
O.~Hope, Jesus College

Originator: Dr A.~Madhavapeddy

8 October 2018
\end{center}

\vspace{5mm}

\textbf{Project Supervisor:} Dr A.~Madhavapeddy

```

```

\textbf{Director of Studies:} Prof C.~Mascolo

\textbf{Project Overseers:} Prof G.~Winskel \& Dr R.~Mortier

% Main document

\section*{Introduction}

Maildir is a widely used format for storing emails both on servers and client machines. Its main benefit is that it uses the

\section*{Starting point}

I have used git for version control of software projects in the past, so I have a basic understanding of some of its more co
MirageOS already has many libraries that will greatly aid the making of an overlay on top of the filesystem and working with
I have previous experience in Standard ML from the Foundations of Computer Science course in Part IA, but all of the differe

\section*{Resources required}

For this project I will use my own Apple Macbook Pro. It runs Mac OS X, has a 2.6 GHz 6 core processor, 512 GB SSD, 16 GB RA

\section*{Work to be done}

The project breaks down into the following sub-projects:

\begin{enumerate}

  \item \textbf{Designing overlay functionality:} Working out exactly what git features will be needed fo the required ove

  \item \textbf{Prototyping the overlay:} Building a version of the overlay to work on linux using the standard git comman

  \item \textbf{Adapting the overlay to MirageOS:} Implementing the overlay on MirageOS using the knowledge gained whilst

  \item \textbf{Adding extra features:} !!!!!!!

  \item \textbf{Evaluation:} Comparing the performance of the indexed Maildir on MirageOS to a standard Maildir implementa

\end{enumerate}

\section*{Success criteria}

The success of the project will be evaluated against:

\begin{enumerate}

  \item Implementation of a working git overlay on top of the filesystem in MirageOS.

  \item The ability to use this overlay to create a strongly consistent index for Maildir

  \item The working implementation of <$insert feature here>$

  \item An evaluation of the performance of the new indexed Maildir against a the same data without the overlay, compared

\end{enumerate}

\section*{Possible extensions}

If I finish both my implementation and evaluation of the project early, then it could be extended by implementing:

\begin{itemize}

  \item Rollback support in case of errors such as accidental deletion

  \item Branch support for processing emails for different platforms. For example a branch for mobile clients which has al

  \item <$INSERT OTHER IDEAS HERE>$

```

```

\end{itemize}

\section*{Timetable}

Planned starting date is 22/10/2018.

\begin{enumerate}

\item \textbf{Michaelmas weeks 2--4 (18th October--31st October):} Learn to use X. Read book Y. Read papers Z.

\item \textbf{Michaelmas weeks 5--6 (1st November--14th November):} Do preliminary test of Q.

\item \textbf{Michaelmas weeks 7--8 (15th November--28th November):} Start implementation of main task A.

\item \textbf{Michaelmas vacation weeks 1--2 (29th November--12th December):} Finish A and start main task B.

\item \textbf{Michaelmas vacation weeks 6--7 (3rd January--16th January):} Finish A and start main task B.

\item \textbf{Lent weeks 1--2 (17th January--30th January):} Write progress report. Generate corpus of
    test examples. Finish task B.

\item \textbf{Lent weeks 3--4 (31st January--5th February):} Run main experiments and achieve working project.

\item \textbf{Lent weeks 5--6 (14th February--27th February):} Second main deliverable here.

\item \textbf{Lent weeks 7--8 (28th February--13th March):} Second main deliverable here.

\item \textbf{Easter vacation:} Extensions and writing dissertation main
    chapters.

\item \textbf{Easter term weeks 1--2 (25th April--8th May):} Further evaluation and complete dissertation.

\item \textbf{Easter term week 3 (9th May--15th May):} Proof reading and then an early submission
    so as to concentrate on examination revision.

\end{enumerate}

\end{document}

```


Appendix B

Makefile

B.1 makefile

B.2 refs.bib

```
@BOOK{Lamport86,  
  TITLE = "{LaTeX} --- a document preparation system --- user's guide  
and reference manual",  
  AUTHOR = "Lamport, L.",  
  PUBLISHER = "Addison-Wesley",  
  YEAR = "1986"}
```

```
@REPORT{Moore95,  
  TITLE = "How to prepare a dissertation in LaTeX",  
  AUTHOR = "Moore, S.W.",  
  YEAR = "1995"}
```


Appendix C

Project Proposal

Computer Science Tripos – Part II – Project Proposal

A strongly consistent index for email using git and
MirageOS.

O. Hope, Jesus College

Originator: Dr A. Madhavapeddy

8 October 2018

Project Supervisor: Dr A. Madhavapeddy

Director of Studies: Prof C. Mascolo

Project Overseers: Prof G. Winskel & Dr R. Mortier

Introduction

Maildir is a widely used format for storing emails both on servers and client machines. Its main benefit is that it uses the filesystem in such a way that client programs do not have to handle locking themselves. The downside of this is that it makes it hard to create a consistent index as we cannot guarantee that the filesystem is in a consistent state when we try to update it. With no index it takes far longer to perform some operations than it would otherwise.

Starting point

I have used git for version control of software projects in the past, so I have a basic understanding of some of its more commonly used features however I will need to learn more about how it actually works and any more advanced features that may be useful. MirageOS already has many libraries that will greatly aid the making of an overlay on top of the filesystem and working with git, so I will be using these to build my project.

I have previous experience in Standard ML from the Foundations of Computer Science course in Part IA, but all of the differences and extra features in OCaml will be new to me.

Resources required

For this project I will use my own Apple Macbook Pro. It runs Mac OS X, has a 2.6 GHz 6 core processor, 512 GB SSD, 16 GB RAM, and a discrete Radeon 560X graphics card. I will backup all of my work on a private Github repository, and will make weekly backups of my entire computer to an external harddrive. Should my laptop suddenly fail, I have a spare and the option to use computers in the Computer Laboratory as well. Alongside my computer, I will take a snapshot of my personal Cambridge email account for testing purposes. I will require no other special resources.

Work to be done

The project breaks down into the following sub-projects:

1. **Designing overlay functionality:** Working out exactly what git features will be needed for the required overlay functionality, how they fit together, and designing a strategy to perform this.
2. **Prototyping the overlay:** Building a version of the overlay to work on linux using the standard git command line tool.
3. **Adapting the overlay to MirageOS:** Implementing the overlay on MirageOS using the knowledge gained whilst prototyping, using pre-existing libraries to manage the use of git.
4. **Adding extra features: !!!!!!!**
5. **Evaluation:** Comparing the performance of the indexed Maildir on MirageOS to a standard Maildir implementation. This will be done according to the metrics specified in the introduction.

Success criteria

The success of the project will be evaluated against:

1. Implementation of a working git overlay on top of the filesystem in MirageOS.
2. The ability to use this overlay to create a strongly consistent index for Maildir
3. The working implementation of <insert feature here>
4. An evaluation of the performance of the new indexed Maildir against the same data without the overlay, compared on a number of axes.

Possible extensions

If I finish both my implementation and evaluation of the project early, then it could be extended by implementing:

- Rollback support in case of errors such as accidental deletion
- Branch support for processing emails for different platforms. For example a branch for mobile clients which has all images removed.
- <INSERT OTHER IDEAS HERE>

Timetable

Planned starting date is 22/10/2018.

1. **Michaelmas weeks 2–4 (18th October–31st October):** Learn to use X. Read book Y. Read papers Z.
2. **Michaelmas weeks 5–6 (1st November–14th November):** Do preliminary test of Q.
3. **Michaelmas weeks 7–8 (15th November–28th November):** Start implementation of main task A.
4. **Michaelmas vacation weeks 1–2 (29th November–12th December):** Finish A and start main task B.
5. **Michaelmas vacation weeks 6–7 (3rd January–16th January):** Finish A and start main task B.
6. **Lent weeks 1–2 (17th January–30th January):** Write progress report. Generate corpus of test examples. Finish task B.
7. **Lent weeks 3–4 (31st January–5th February):** Run main experiments and achieve working project.
8. **Lent weeks 5–6 (14th February–27th February):** Second main deliverable here.
9. **Lent weeks 7–8 (28th February–13th March):** Second main deliverable here.
10. **Easter vacation:** Extensions and writing dissertation main chapters.
11. **Easter term weeks 1–2 (25th April–8th May):** Further evaluation and complete dissertation.
12. **Easter term week 3 (9th May–15th May):** Proof reading and then an early submission so as to concentrate on examination revision.