

ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA VẬT LÝ - VẬT LÝ KỸ THUẬT
CHUYÊN NGÀNH VẬT LÝ TIN HỌC

—oOo—

KHOÁ LUẬN TỐT NGHIỆP

Đề tài:

**GIẢI TÍCH SỐ TÍCH PHÂN
MONTE CARLO NHIỀU LỚP
BẰNG GIEO ĐIỂM QUAN
TRỌNG**

SVTH: Huỳnh Thị Hạ Vy

CBHD: TS. Nguyễn Chí Linh

TP. HỒ CHÍ MINH - 2019

ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA VẬT LÝ - VẬT LÝ KỸ THUẬT
CHUYÊN NGÀNH VẬT LÝ TIN HỌC

—————oOo—————

KHOÁ LUẬN TỐT NGHIỆP

Đề tài:

GIẢI TÍCH SỐ TÍCH PHÂN
MONTE CARLO NHIỀU LỚP
BẰNG GIEO ĐIỂM QUAN
TRỌNG

SVTH: Huỳnh Thị Hạ Vy

CBHD: TS. Nguyễn Chí Linh

—————
TP. HỒ CHÍ MINH - 2019

Lời cảm ơn

Đầu tiên, con xin gửi lời biết ơn đến mẹ, người đã thay thế vai trò người cha đã mất, cáng đáng cả gia đình và nuôi dưỡng con nên người, con cũng xin cảm ơn dì Chính, người mà còn vẫn luôn coi như người mẹ thứ hai, chăm sóc con từng miếng ăn, giấc ngủ và luôn coi con như con ruột của mình, công ơn của hai mẹ dành cho con không từ ngữ nào mà diễn tả được.

Em xin cảm ơn các thầy cô khoa Vật Lý - Vật Lý Kỹ Thuật, đã tận tâm truyền đạt kiến thức cho em trong những năm đầu đại học. Em xin chân thành cảm ơn thầy cô của Bộ môn Vật Lý Tin Học, đã xây dựng bộ môn với các trang thiết bị hiện đại và sự nhiệt tình, thân thiện của các thầy cô, giúp em có thể thoải mái học tập, nghiên cứu mà không cảm thấy căng thẳng, áp lực. Những lời chỉ bảo của thầy cô đã cho em những kiến thức cần thiết và quý báu cho định hướng của mình.

Và em cũng xin gửi lời cảm ơn tới thầy TS. Nguyễn Chí Linh, đã giới thiệu và hướng em vào đề tài này khi em không xác định được hướng đi cho mình, thầy cũng dành thời gian đọc, chỉnh sửa và góp ý cho bài báo cáo này được hoàn thiện hơn. Đồng thời tôi cũng muốn cảm ơn những người bạn ở Vật Lý Lý Thuyết, đã dành thời gian với tôi trong những ngày mới bước vào chuyên ngành, thông qua những buổi nói chuyện đó, tôi mới lần đầu biết đến nền tảng LaTeX.

Và cuối cùng, tôi xin cảm ơn những người bạn, những người đàn em đã cùng đồng hành với tôi trong suốt bốn năm trên giảng đường Đại học, cảm ơn vì những khoảng khắc trò chuyện vui vẻ giúp giải toả áp lực đã trở thành một phần kỉ niệm của đời sinh viên.

TP. Hồ Chí Minh, tháng 1 năm 2018.

Trịnh Tích Thiện

Mục lục

Các kí hiệu viết tắt	iii
Danh sách hình vẽ	iv
Danh sách bảng	vi
Lời giới thiệu	1
1 Kiến thức cơ sở xác suất và thống kê	3
1.1 Biến ngẫu nhiên và xác suất	3
1.2 Phân phối xác suất	4
1.2.1 Hàm khối xác suất của biến rời rạc (PMF)	4
1.2.2 Hàm mật độ xác suất của biến liên tục (PDF)	4
2 Soạn thảo văn bản trong L^AT_EX	5
2.1 Cài đặt L ^A T _E X	5
2.1.1 Cài đặt L ^A T _E X trên Linux	5
2.1.2 Cài đặt L ^A T _E X trên Windows	7
2.2 Hướng dẫn sơ lược	14
2.2.1 Đặc điểm của L ^A T _E X	14
2.2.2 Soạn thảo văn bản	17
3 Thiết kế định dạng văn bản riêng trong L^AT_EX	29
3.1 Tổng quan về gói và lớp	29
3.2 Giới thiệu lớp vlththesis	30
3.3 Cấu trúc của vlththesis	30
3.3.1 Các khai báo ban đầu	30
3.3.2 Khai báo gói và thiết lập	33
3.3.3 Các câu lệnh của vlththesis	52
4 Kết luận và hướng phát triển	56
4.1 Kết luận	56
4.2 Hướng phát triển của đề tài	56
A Định nghĩa đầy đủ của câu lệnh	59

B Trích đoạn tập tin đầu vào \LaTeX của đề tài	64
Tài liệu tham khảo	67

Các kí hiệu viết tắt

AMS	American Mathematical Society
CTAN	Comprehensive T _E X Archive Network
ISBN	International Standard Book Number
LoF	List of Figures
LoT	List of Tables
LPPL	LaTeX Project Public License
ToC	Table of Contents
WYSIWYG	What you see is what you get

Danh sách hình vẽ

Hình 1.1	Các sự kiện có thể xuất hiện khi gieo đồng xu	3
Hình 1.2	Các sự kiện có thể xuất hiện khi gieo súc sắc	4
Hình 2.1	Kết quả có được sau khi nhập câu lệnh cài Kile	5
Hình 2.2	Giao diện trang tải MiKTeX	7
Hình 2.3	Chọn tải các gói từ mạng	8
Hình 2.4	Chọn gói muốn tải	8
Hình 2.5	Chọn nguồn tải	9
Hình 2.6	Chỉ định đường dẫn tới thư mục chứa gói đã tải về	9
Hình 2.7	Quá trình download đang được tiến hành	10
Hình 2.8	Chọn tùy chọn cài đặt MiKTeX	10
Hình 2.9	Chọn gói MiKTeX cần cài đặt	11
Hình 2.10	Chọn quyền truy cập MiKTeX	11
Hình 2.11	Trỏ đường dẫn tới thư mục chứa các gói	12
Hình 2.12	Chọn thư mục cài đặt	12
Hình 2.13	Chọn khổ giấy mặc định và tự động tải	13
Hình 2.14	Quá trình cài đặt đang được tiến hành	13
Hình 2.15	Kết quả sau khi chạy tập tin đầu vào	16
Hình 2.16	Ví dụ một văn bản LaTeX đơn giản	18
Hình 2.17	Văn bản đầu ra được định dạng bằng các lệnh cơ bản	21
Hình 2.18	Ví dụ cho câu lệnh <code>\emph</code>	22
Hình 2.19	Ví dụ xuống dòng	23
Hình 2.20	Ví dụ canh vị trí	24
Hình 2.21	Ví dụ thêm hình trong văn bản	25
Hình 2.22	Ví dụ tạo bảng	26
Hình 2.23	Đánh dấu chương và đề mục	28
Hình 3.1	So sánh đề mục của hai lớp văn bản	37
Hình 3.2	So sánh đề mục (có đánh số) của hai lớp văn bản	38
Hình 3.3	So sánh chú thích của hai lớp văn bản	39
Hình 3.4	So sánh tiêu đề danh sách hình vẽ của hai lớp văn bản	48
Hình 3.5	So sánh tiêu đề danh sách bảng và mục lục của hai lớp văn bản	48
Hình 3.6	Định dạng mặc định của LaTeX dành cho thành phần danh sách	49

Hình 3.7	Nội dung và nhãn bị chồng nhau sau câu lệnh bổ sung	50
Hình 3.8	Kết quả mong muốn của lớp <code>vlththesis.cls</code>	50
Hình 3.9	Ví dụ gói <code>glossaries</code>	51

Danh sách bảng

Bảng 2.1	Các câu lệnh chương mục và thứ bậc	28
-----------------	--	----

Lời giới thiệu

Ngày nay, ngoài các trình soạn thảo văn bản phổ biến, LaTeX cũng là một sự lựa chọn dành cho người soạn thảo được tạo ra với triết lý hoàn toàn khác biệt so với các trình hiện hành. Nhận thấy hạn chế của chất lượng in ấn ở những năm 1970, và việc người dùng tốn quá nhiều thời gian để định dạng thay vì tập trung soạn thảo, Donald E.Knuth đã phát triển hệ thống TeX, và từ đó, Leslie Lamport xây dựng thành LaTeX, với mục đích giúp người dùng sử dụng câu lệnh để việc thiết kế văn bản được thực hiện một cách tự động bởi hệ thống.

Tuy xuất hiện đã lâu nhưng do không có tính trực quan vốn có của các trình soạn thảo văn bản thông thường cũng như đòi hỏi người sử dụng có khái niệm cơ bản, về ngôn ngữ đánh dấu (markup language), cộng thêm việc nền tảng này chỉ lưu hành trong giới học thuật, nên LaTeX vẫn chưa thực sự phổ biến đến những người dùng phổ thông (mặc dù đối tượng sử dụng ngày càng đa dạng).

Nhận thấy LaTeX thích hợp để tạo các văn bản có quy chuẩn rõ ràng, đồng thời nền tảng cho phép người dùng thiết kế bố cục và kiểu văn bản cho riêng mình, đề tài này đã ra đời nhằm mục đích thiết kế, xây dựng một mẫu báo cáo khoá luận chuẩn trên nền LaTeX, định nghĩa các câu lệnh mới để hỗ trợ những người dùng sau này có thể dễ dàng định dạng các báo cáo khoá luận mà không tốn nhiều thời gian vào việc thiết kế, canh chỉnh, thay vào đó tập trung hơn vào nội dung và thành phần văn bản của mình, kế thừa đúng với tinh thần của những người sáng tạo ra LaTeX.

Tài liệu về LaTeX tuy đa dạng, nhưng lại có tính chuyên môn, đòi hỏi thời gian tìm hiểu và tổng hợp những tài liệu thật sự cần thiết, nhưng cũng nhờ đó, tôi đã có thêm kĩ năng đọc hiểu, tìm kiếm thông tin, đồng thời hiểu thêm được các khái niệm, thao tác lập trình với macro, cũng như tiếp cận và biết thêm được nhiều thủ thuật soạn thảo, trình bày văn bản theo ý mình sử dụng LaTeX, và đó là những lý do tôi chọn đề tài này. Thông qua đề tài, ngoài việc xây dựng thành công một mẫu khoá luận, tôi cũng muốn phổ biến sự tiện lợi trong việc soạn thảo các văn bản khoa học của LaTeX đến nhiều người hơn bằng việc giới thiệu, đưa ra những hướng dẫn cơ bản và tổng hợp những nguồn tham khảo tin cậy cho hệ thống LaTeX này.

Báo cáo đề tài gồm bốn chương chính như sau:

- **Chương 1: Tổng quan về LaTeX.** Giới thiệu khái niệm của LaTeX và lịch sử hình thành của hệ thống, đồng thời giới thiệu sơ lược về trình soạn thảo hỗ trợ LaTeX.
- **Chương 2: Soạn thảo văn bản trong LaTeX.** Hướng dẫn cách tải và cài đặt nền tảng LaTeX trên hai hệ điều hành Windows và Linux, đồng thời đưa ra những hướng dẫn cơ bản về cách soạn thảo văn bản bằng LaTeX, các khái niệm, thuật ngữ và câu lệnh cần nắm để dễ dàng hiểu được các tài liệu hướng dẫn LaTeX.
- **Chương 3: Thiết kế định dạng văn bản riêng trong LaTeX.** Sẽ tập trung vào cách thức thiết kế các định dạng văn bản riêng trong LaTeX, từ đó tiến tới thiết kế bài báo cáo, luận văn, sau đó phân tích quy trình tạo và cấu trúc của tập tin (file) sản phẩm đề tài.
- **Chương 4: Kết luận và hướng phát triển.** Đưa ra kết luận về kết quả thu được của đề tài này và đánh giá hướng phát triển của thành phẩm.

CHƯƠNG 1

Kiến thức cơ sở xác suất và thống kê

1.1 Biến ngẫu nhiên và xác suất

Biến ngẫu nhiên là một hàm ánh xạ với đặc điểm nó gán một giá trị bằng số cho kết quả đầu ra của một phép thử ngẫu nhiên.

$$\mathbf{X}(\omega) = x \quad (1.1)$$

với ω là đại diện cho đầu ra của một thực nghiệm, x là một số thực (hay sự kiện), \mathbf{X} là hàm ánh xạ (hay biến ngẫu nhiên). Tính ngẫu nhiên được thể hiện ở tham số đầu vào ω . Điều này dẫn đến đầu ra của hàm là ngẫu nhiên.

Đây chưa phải là định nghĩa đầy đủ của một biến ngẫu nhiên. Khái niệm khác liên quan đến định nghĩa của một biến ngẫu nhiên là khái niệm xác suất.

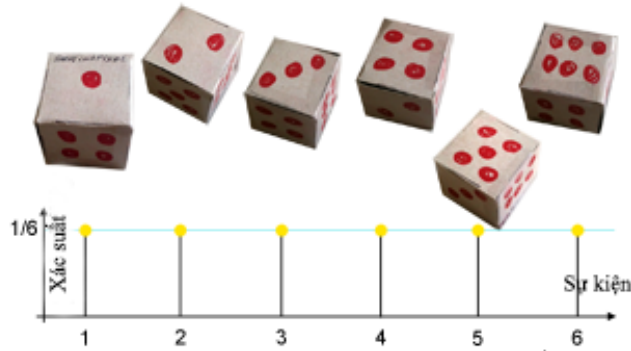
Xét một số thí dụ sau.

- Gieo một đồng xu trên mặt phẳng, đây là một phép thử. Kết quả có thể xảy ra là "sấp" hoặc "ngửa". Như vậy xác suất cho sự xuất hiện của "sấp" và "ngửa" lần lượt là $\frac{1}{2}$ và $\frac{1}{2}$



Hình 1.1: Các sự kiện có thể xuất hiện khi gieo đồng xu

- Gieo một con súc sắc, đây là một phép thử. Kết quả có thể xảy ra là "Xuất hiện mặt k chấm" tương ứng với $k = 1, 2, 3, \dots, 6$. Xác suất cho mỗi sự kiện "Xuất hiện k chấm" đều là $\frac{1}{6}$.



Hình 1.2: Các sự kiện có thể xuất hiện khi gieo súc sắc

Tổng quát, nếu một phép thử tạo ra n sự kiện khác nhau và khả năng xảy ra như nhau thì xác suất của mỗi sự kiện là $\frac{1}{n}$ chúng ta có thể nói rằng nếu kết quả của một quá trình nào đó phải là một trong n kết quả khác nhau.

1.2 Phân phối xác suất

1.2.1 Hàm khối xác suất của biến rời rạc (PMF)

1.2.2 Hàm mật độ xác suất của biến liên tục (PDF)

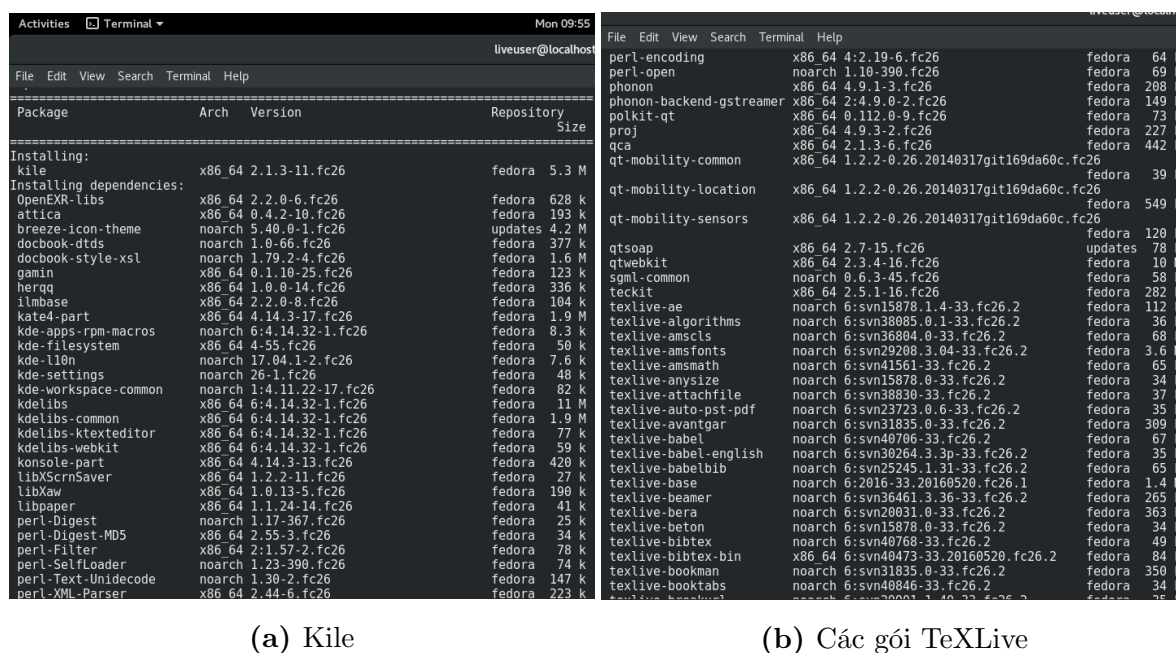
CHƯƠNG 2

Soạn thảo văn bản trong \LaTeX

2.1 Cài đặt \LaTeX

2.1.1 Cài đặt \LaTeX trên Linux

LateX có thể dễ dàng cài đặt trên Linux bằng cách cài trình soạn thảo hỗ trợ nó. Thông thường, một số hệ điều hành nhân Linux sẽ tự động tìm kiếm và cài đặt bổ sung TeXLive nhằm khởi chạy được trình soạn thảo đó. Ví dụ ở hình 2.1 dưới đây sử dụng hệ điều hành Fedora.



Hình 2.1: Kết quả có được sau khi nhập câu lệnh cài Kile

Sau câu lệnh `sudo dnf install kile`, ta có thể thấy, ngoài Kile, hệ thống tự động tìm kiếm và tải về TeXLive và các gói chương trình khác *vừa đủ* để khởi chạy trình soạn thảo trên.

Tuy nhiên, sẽ có trường hợp ta muốn đích thân tải hay chỉ muốn tải TeXLive (bản có LaTeX) hoặc hệ điều hành không tự động tải về. Ta có thể thực hiện như sau:

- Đối với các hệ điều hành Debian, Ubuntu,... ta nhập câu lệnh sau vào terminal:

```
# apt-get install texlive texlive-base
```

hoặc

```
# apt-get install texlive-full
```

Câu lệnh `texlive-full` sẽ tải hết tất cả các gói của LaTeX.

- Đối với các hệ điều hành RedHat, CentOS, Fedora,... ta nhập:

```
# yum install texlive texlive-latex
```

Ta có thể sẽ cần quyền admin, trường hợp đó chỉ cần thêm `sudo` vào trước câu lệnh và nhập mật khẩu admin khi được hỏi. Đối với các bản Fedora từ 18 trở đi, nên sử dụng `dnf` thay cho `yum`.

Sau đó, nếu muốn, ta có thể tiến hành cài đặt trình soạn thảo hỗ trợ LaTeX như bình thường. Mặc định TeXLive sau khi được cài đặt sẽ nằm ở đường dẫn `/usr/share/texlive`, và các gói của LaTeX sẽ nằm ở `/usr/share/texlive/texmf-dist/tex/latex`.

Người dùng còn có thể nhập câu lệnh `yum install texlive-latex-doc` (CentOS, RedHat,...) hoặc `apt-get install texlive-latex-doc` (Ubuntu, Debian,...), để tải về bộ tài liệu thông tin và hướng dẫn cho LaTeX và các gói, lớp tiêu chuẩn. Bộ tài liệu mặc định nằm ở `usr/share/texlive/texmf-dist/doc/latex/base`.

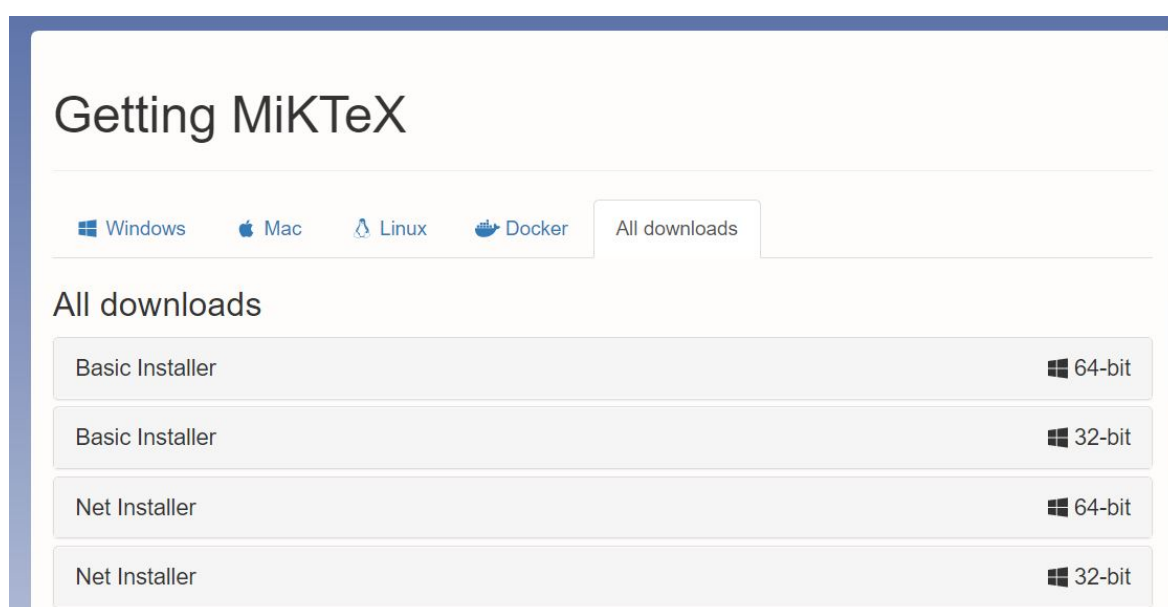
Tương tự, để tải và cài đặt các gói LaTeX (khi được hệ thống yêu cầu để chạy tập tin `.tex`), người dùng cũng có thể sử dụng `yum` hoặc `apt-get` kèm theo `install texlive-<tên gói>`, Linux sẽ tự động tìm kiếm, tải (nếu có) và cài đặt gói mở rộng trên.

2.1.2 Cài đặt $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ trên Windows

Trên nền Windows, ta có thể chọn TeXLive làm gói phân phối LaTeX, tuy nhiên, MiKTeX hỗ trợ, đồng bộ tốt hơn cho Windows và cũng miễn phí như TeXLive, nên đối với hệ điều hành này, ta sẽ chọn MiKTeX.

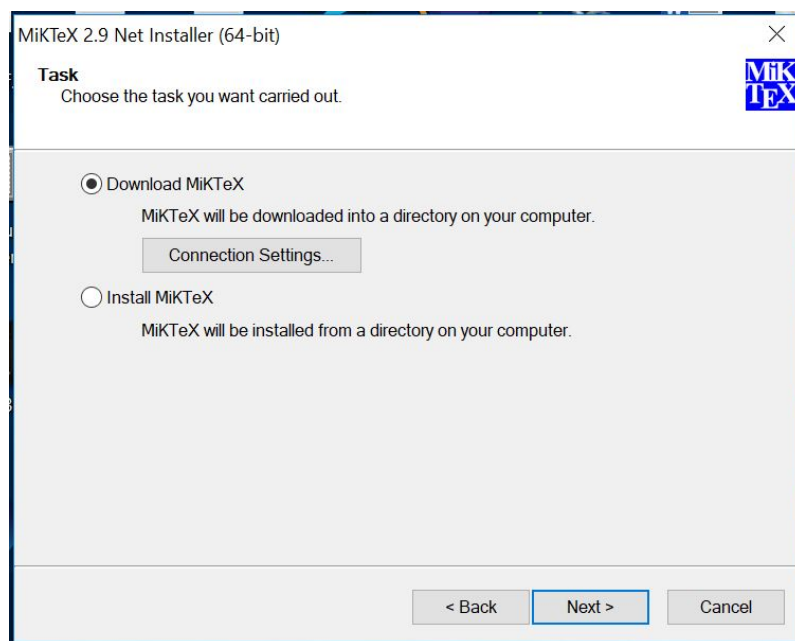
MiKTeX có thể được cài đặt qua các bước sau:

- *Bước 1:* Truy cập trang web <https://miktex.org/download>, để tải trình cài đặt (installer) cho MiKTeX. Click vào tab **All downloads** để chọn bản cài đặt phù hợp với hệ điều hành đang sử dụng. Nên chọn bản Net installer để có đầy đủ chức năng.

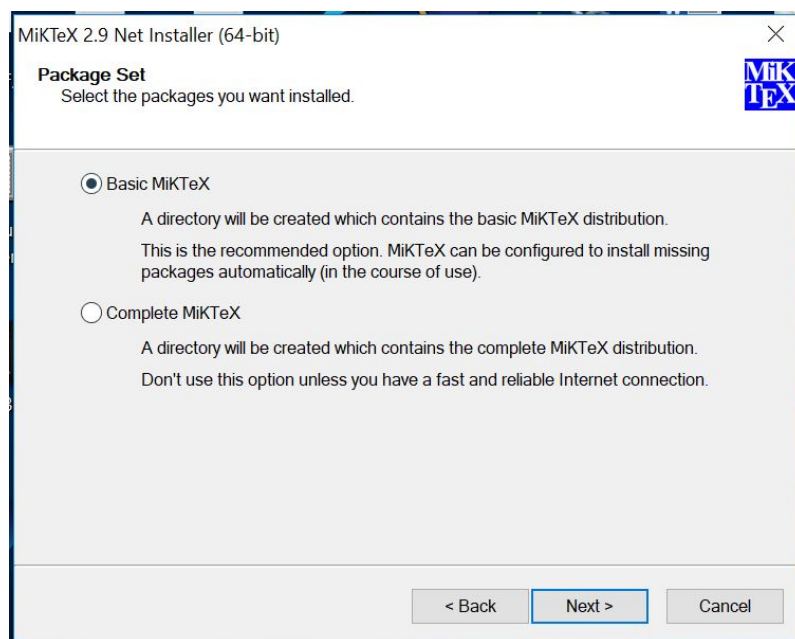


Hình 2.2: Giao diện trang tải MiKTeX

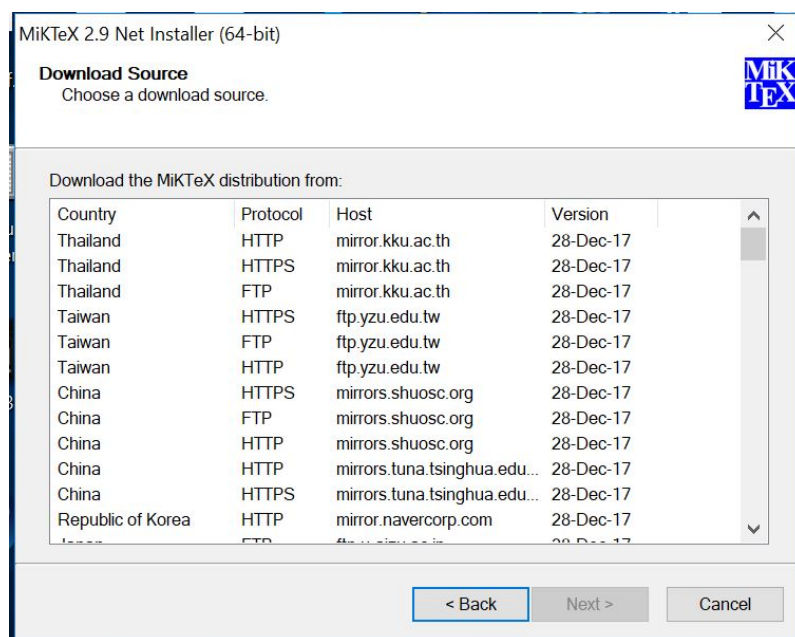
- *Bước 2:* Mở installer và tiến hành cài đặt, sau khi chấp nhận điều khoản sử dụng, ở bước tiếp theo, có hai lựa chọn cài đặt gói có sẵn trong máy (Install MikTeX), hay tải (download) các gói từ mạng (Download MikTeX). Do đây là lần cài đặt đầu tiên, ta nên chọn tải gói về như minh hoạ ở hình 2.3.

**Hình 2.3:** Chọn tải các gói từ mạng

- *Bước 3:* Ta có hai tùy chọn Basic \LaTeX (các gói cơ bản của \LaTeX) hoặc Complete \LaTeX (tất cả các gói chính thức của \LaTeX), tùy theo tốc độ và nhu cầu sử dụng mà ta đưa ra lựa chọn, tuy nhiên, nên lựa chọn gói Basic để việc download được tiến hành nhanh chóng. MiKTeX có cơ chế tự động tải các gói còn thiếu, nếu phát hiện trong tập tin văn bản \LaTeX sử dụng gói không có sẵn trong máy.

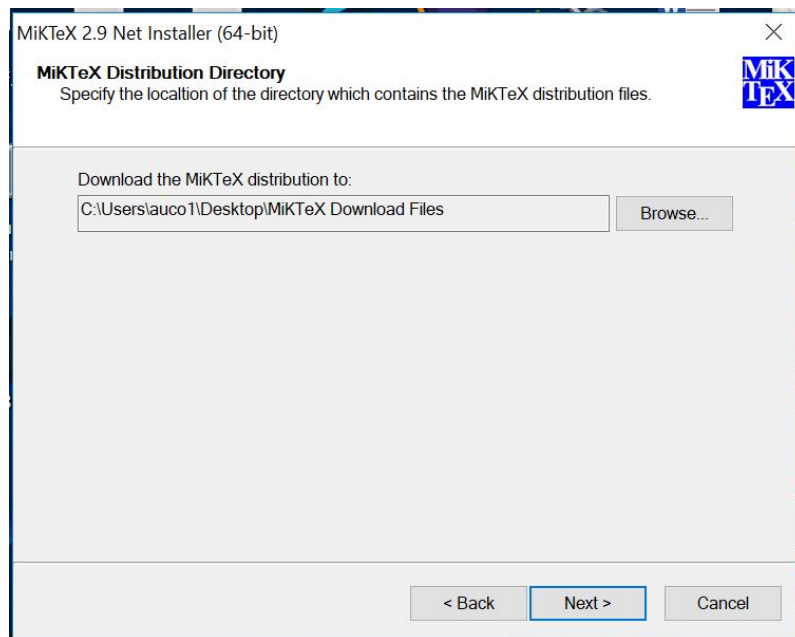
**Hình 2.4:** Chọn gói muốn tải

- *Bước 4:* Sau khi chọn loại gói, tiếp theo sẽ là danh sách các nguồn cung cấp gói, ta nên chọn nguồn gần nhất và giao thức FTP để đảm bảo tốc độ download.



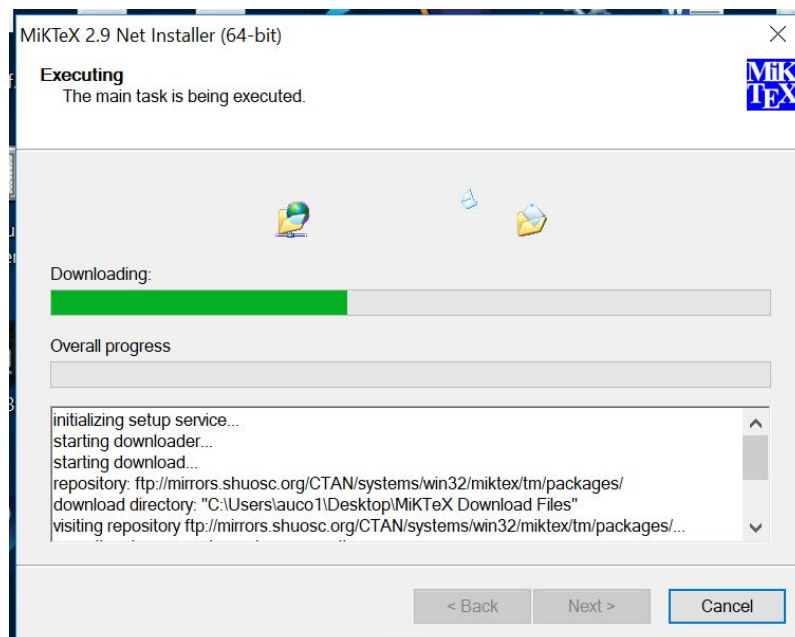
Hình 2.5: Chọn nguồn tải

- *Bước 5:* Ở bước này ta chỉ định đường dẫn tới thư mục mà ta muốn đặt các gói tải về.



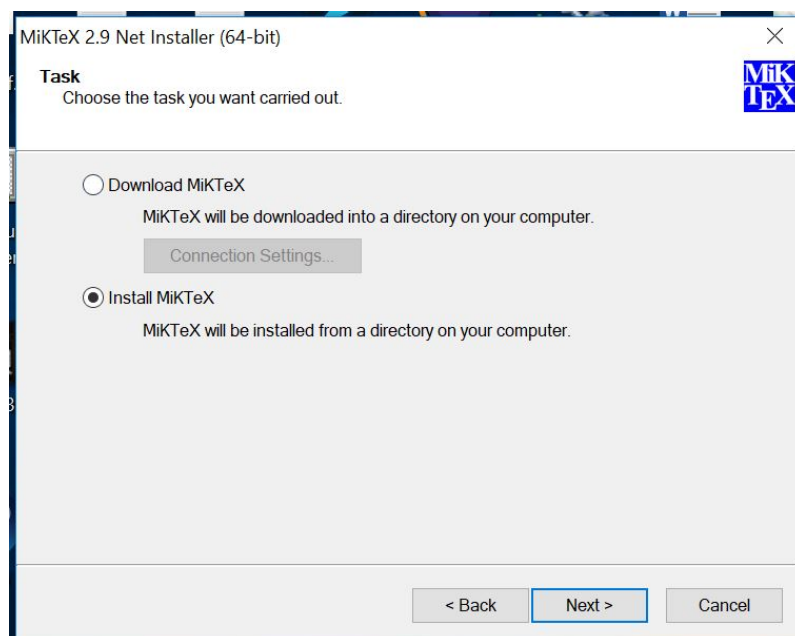
Hình 2.6: Chỉ định đường dẫn tới thư mục chứa gói đã tải về

Sau đó, chọn **Next**, bước kế tiếp là xác nhận thông tin, nếu không có gì cần thay đổi, ta bấm **Start** để tiến hành tải gói.



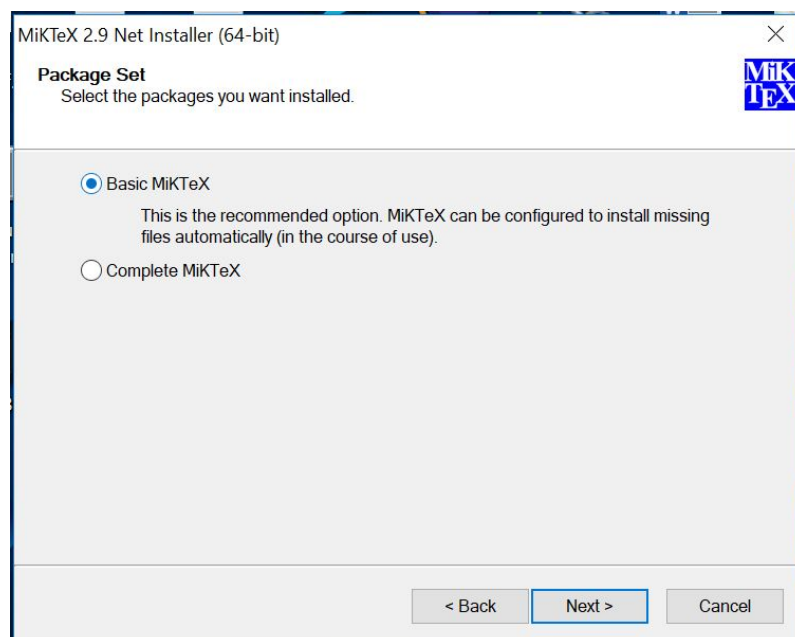
Hình 2.7: Quá trình download đang được tiến hành

- *Bước 6*: Lặp lại bước 2, nhưng lần này chọn tùy chọn **Install MiKTeX**.



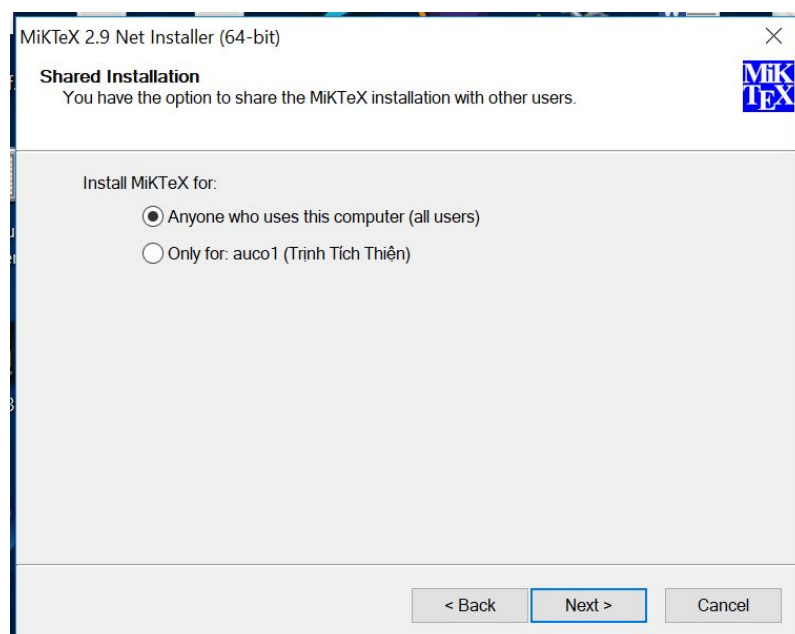
Hình 2.8: Chọn tùy chọn cài đặt MiKTeX

- *Bước 7:* Chọn gói cài đặt ứng với gói mà ta đã tải về ở bước 3.



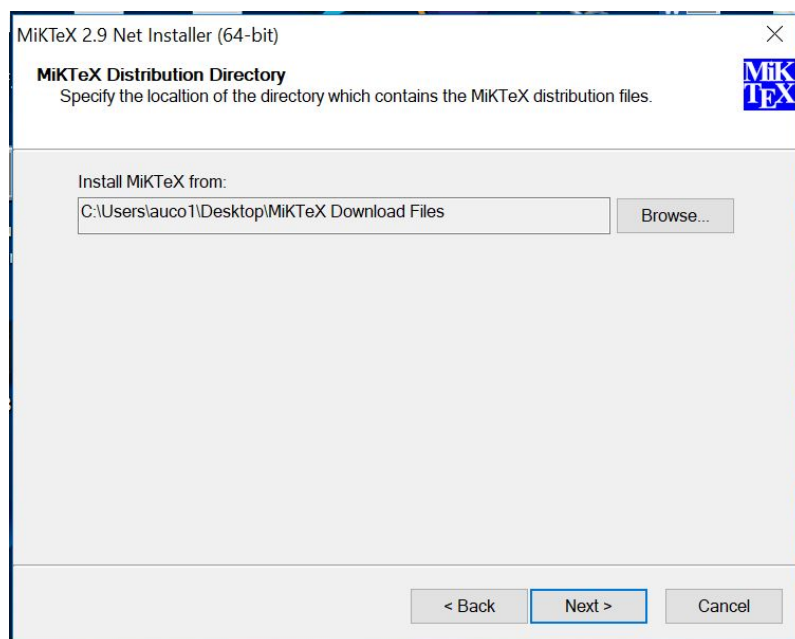
Hình 2.9: Chọn gói MiKTeX cần cài đặt

Sau đó, chọn quyền truy cập MiKTeX cho phép ai cũng sử dụng được (Anyone...) hoặc chỉ tài khoản hiện tại (Only for:...).



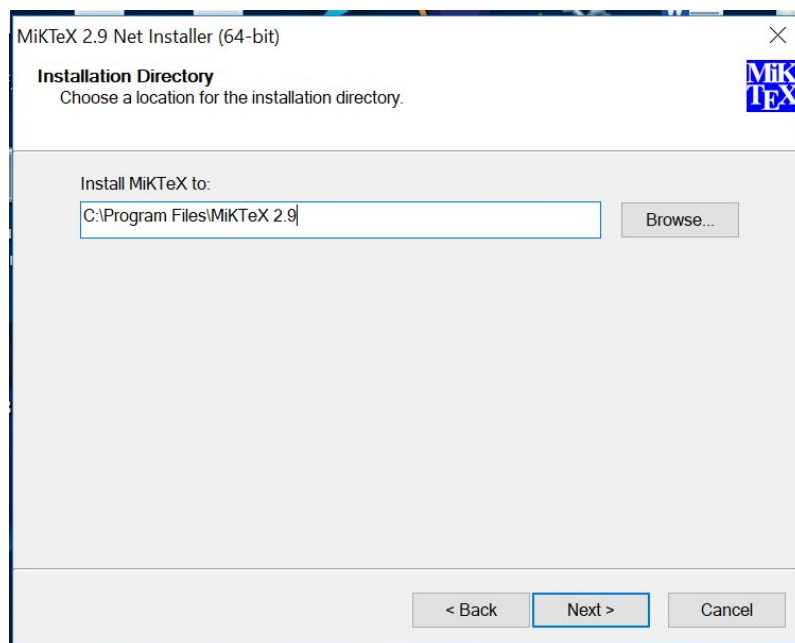
Hình 2.10: Chọn quyền truy cập MiKTeX

- *Bước 8:* Tiếp theo, trỏ đường dẫn tới thư mục mà ta đã tải các gói về.



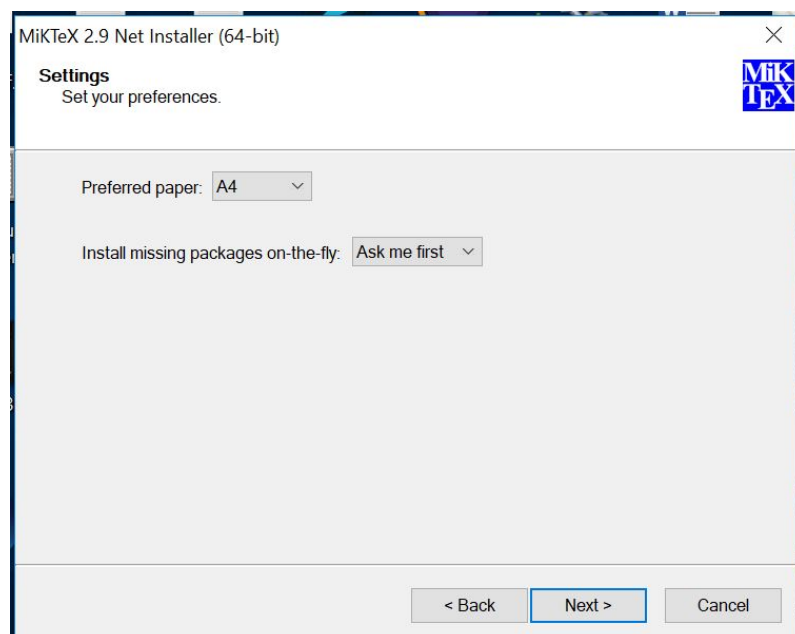
Hình 2.11: Trỏ đường dẫn tới thư mục chứa các gói

Sau đó, chọn đường dẫn tới thư mục ta muốn cài đặt MiKTeX.



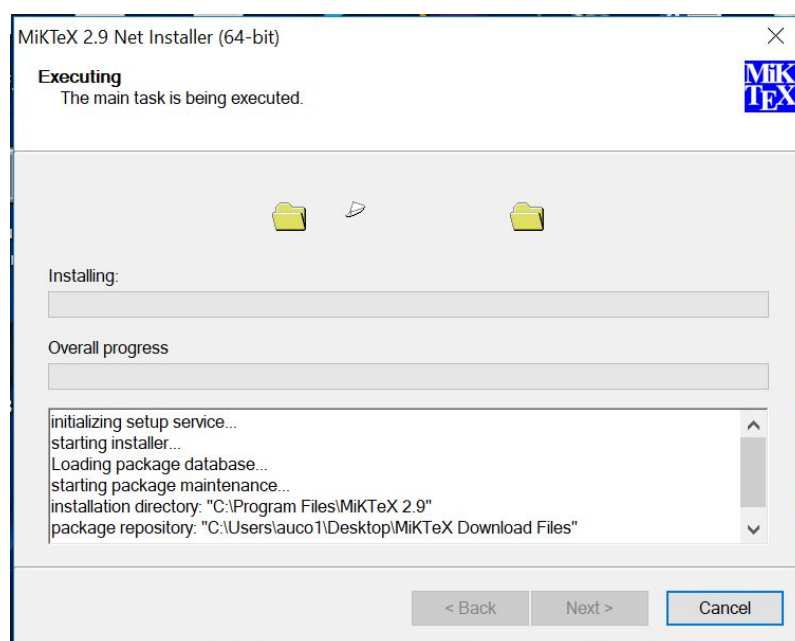
Hình 2.12: Chọn thư mục cài đặt

- *Bước 9:* Ở bước này ta chọn khổ giấy mặc định (ở đây chọn A4) và các tùy chọn cho phép (Yes) hay không cho phép (No) hệ thống tự động tải g còn thiếu hoặc hỏi ý kiến trước khi tải (Ask me first).



Hình 2.13: Chọn khổ giấy mặc định và tự động tải

- *Bước 10:* Tối bước xác định thông tin, ta bấm **Start** để tiến hành cài đặt nếu không cần phải thay đổi gì.



Hình 2.14: Quá trình cài đặt đang được tiến hành

Sau khi cài đặt hoàn tất, thông báo cài đặt sẽ hiện lên, ta chọn **Close** để kết thúc quá trình cài đặt.

Khi đã hoàn tất cài đặt MiKTeX, ta cần chọn trình soạn thảo LaTeX hỗ trợ Windows, ngoài các trình bản quyền, ta cũng có các trình mã nguồn mở, miễn phí. Trong các trình soạn thảo LaTeX mã nguồn mở cho Windows, TeXstudio được sử dụng thường xuyên nhất, người dùng quan tâm có thể tham khảo và tải trình soạn thảo này tại <https://www.texstudio.org/>.

2.2 Hướng dẫn sơ lược

2.2.1 Đặc điểm của L^AT_EX

Khác với các trình soạn thảo hay xử lý văn bản ngày nay, cho phép ta được nhìn thấy hình thức trình bày của văn bản trong quá trình soạn thảo (hay còn được gọi là “**WYSIWYG**”), với LaTeX, ta để việc thiết kế, định dạng cho các macro bằng việc sử dụng các câu lệnh để đánh dấu (hay mô tả, định danh) ý nghĩa, mục đích của nội dung mà ta soạn thảo, cũng giống như HTML.

Sử dụng các câu lệnh của LaTeX, ta viết ra một tập tin chứa nội dung và các “mô tả” đó bằng các trình soạn thảo hỗ trợ, các tập tin đó có tên mở rộng (extension) là `.tex` hay còn được gọi là tập tin đầu vào LaTeX (input file). Các tập tin đầu vào sau đó sẽ được biên dịch ra mã TeX bởi các trình soạn thảo, sử dụng các macro và định nghĩa có trong gói phân phối và xây dựng/chạy (build) thành tập tin văn bản (`dvi`, `pdf`,...) dùng để đọc và in. Ta lấy ví dụ một đoạn văn bản sau:

Tiêu đề của văn bản này
Nguyễn Văn A
Tháng 9 2015

Hello World!

Với các trình xử lý văn bản **WYSIWYG**, trước tiên ta sẽ chọn font, cỡ chữ,... cũng như các hiệu ứng khác nhau nhằm gợi ý người đọc ý nghĩa của nội dung (ví dụ tên người thì in nghiêng, tiêu đề cỡ chữ to, in đậm, căng lề giữa,...). Đối với LaTeX, ta sẽ trình bày trong tập tin đầu vào như sau:

```
\documentclass{article}
\usepackage[utf8]{inputenc}
\usepackage[vietnamese]{babel}
\title{Tiêu đề của văn bản này}
\author{Nguyễn Văn A}
\date{Tháng 9 2015}
\begin{document}
  \maketitle
  Hello World!
\end{document}
```

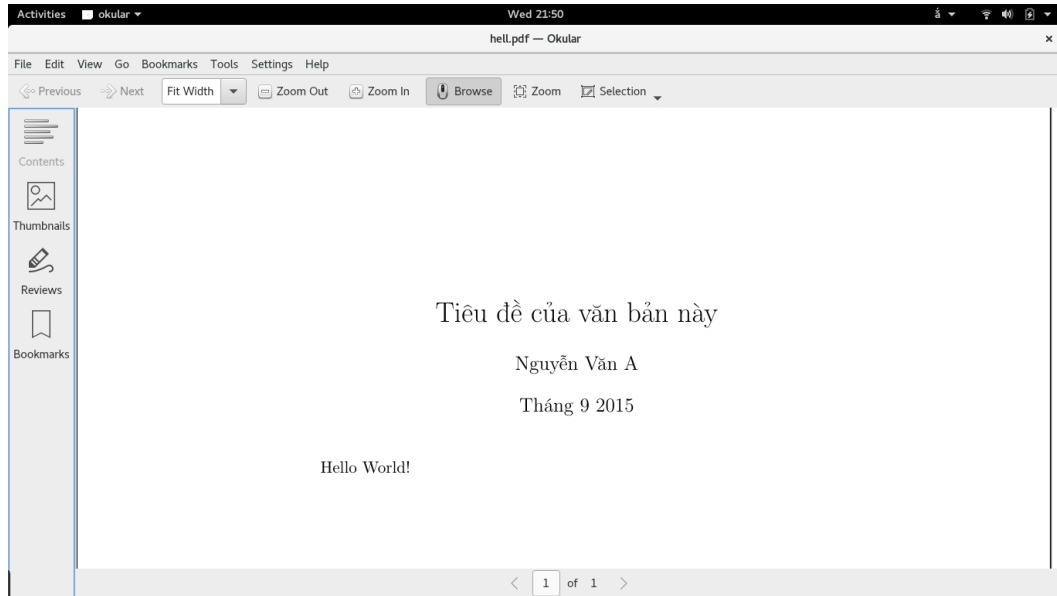
Bằng việc khai báo như trên, ta như ngụ ý với LaTeX rằng:

- Văn bản này là một bài báo (article).
- Sử dụng gói `inputenc` với tùy chọn (option) **utf8** để xử lý các kí tự UTF-8¹.
- Sử dụng gói `babel` với tùy chọn `vietnamese` hỗ trợ định dạng tiếng Việt.
- Tiêu đề của nó là “Tiêu đề của văn bản này”.
- Người viết là “Nguyễn Văn A”.
- Văn bản được viết vào tháng 9, 2015.
- Nội dung văn bản bao gồm tiêu đề và Hello World!

Dựa vào những mô tả trên, hệ thống biên dịch sẽ tìm kiếm định nghĩa của các câu lệnh trên trong tập tin của lớp (class) **article** (tức lớp LaTeX dùng để viết bài báo), lớp là một gói tập tin đặc biệt của LaTeX, định nghĩa các câu lệnh cơ bản và cần thiết cho một văn bản. Ngoài ra, hệ thống còn tìm kiếm trong gói `babel` và `inputenc` cách định dạng và mã hoá tiếng Việt. Sau khi chạy (build) tập tin `.tex` trên ta được văn bản như hình 2.15.

LaTeX tự động đưa ra những định dạng khác nhau cho từng đối tượng của văn bản mà không cần ta đích thân làm công việc đó. Tất cả những gì cần làm là nhập nội dung và “nói” cho LaTeX biết nội dung đó thật ra là gì. Là tiêu đề, hay nội dung bình thường của bài báo.

¹UTF-8 là phương thức mã hoá phổ biến để miêu tả bảng mã Unicode trong bộ nhớ



Hình 2.15: Kết quả sau khi chạy tập tin đầu vào

Như đã đề cập ở 1.3, LaTeX và TeX đã giới thiệu khái niệm logical design, tức người dùng chỉ nên tập trung vào cấu trúc logic và nội dung văn bản hơn là định dạng, trái với visual design (hay còn được gọi là thiết kế hình ảnh) thường thấy ở các trình soạn thảo **WYSIWYG**. Đối với các văn bản ngắn, cần đòi hỏi nhiều hiệu ứng, các trình soạn thảo **WYSIWYG** là thích hợp, nhưng khi đối diện với các văn báo cáo khoa học, với nhiều kí hiệu và phương trình phức tạp, các trình soạn thảo thông thường sẽ khiến người dùng tốn nhiều công sức trong việc tìm kiếm, bổ sung, định dạng các kí hiệu và phương trình đó.

Bên cạnh các kí hiệu khoa học, LaTeX còn hỗ trợ các câu lệnh và gói² cho phép người dùng tạo siêu liên kết (hyperlink) đến nội bộ văn bản hoặc bên ngoài, tự động tổng hợp và tạo danh mục hình ảnh, bảng, mục lục chỉ với một câu lệnh duy nhất, tạo, trình bày, kiểm soát danh sách tài liệu tham khảo (reference hay bibliography), và giúp bố cục trình bày được đồng nhất xuyên suốt văn bản.

Ta xét một ví dụ sau:

$$\mathcal{L}_{quarks} = [i\bar{\psi}_r\gamma^\mu\partial_\mu\psi_r - m\bar{\psi}_r\psi_r] + [i\bar{\psi}_b\gamma^\mu\partial_\mu\psi_b - m\bar{\psi}_b\psi_b] + [i\bar{\psi}_g\gamma^\mu\partial_\mu\psi_g - m\bar{\psi}_g\psi_g]$$

Đối với các trình soạn thảo thông thường, ta sẽ phải tốn nhiều thời gian chỉ để tìm kiếm các kí tự, canh chỉnh các dấu gạch (—) sao cho phù hợp, hoặc để nhanh gọn, có thể chụp phương trình trên ở đâu đó và bổ sung vào văn bản như một ảnh kèm vào

²Một số gói không có sẵn trong gói phân phối và phải được tải bổ sung

nội dung nhưng khi đó ta cũng không thể chỉnh sửa được. Để hiển thị phương trình trên bằng LaTeX, ta chỉ cần nhập:

```
\mathcal{L}_{\text{quarks}} = \left[ i\bar{\psi}_r \gamma^{\mu} \partial_{\mu} \psi_r - m\bar{\psi}_r \psi_r \right] + \left[ i\bar{\psi}_b \gamma^{\mu} \partial_{\mu} \psi_b - m\bar{\psi}_b \psi_b \right] + \left[ i\bar{\psi}_g \gamma^{\mu} \partial_{\mu} \psi_g - m\bar{\psi}_g \psi_g \right]
```

Đúng là khi nhìn vào lệnh LaTeX sẽ thấy phức tạp hơn, nhưng rõ ràng việc này giúp ta dễ dàng chỉnh sửa nếu có sai sót cũng như không cần phải bỏ công định dạng. Khi ta biết rõ được chức năng của từng câu lệnh trên, việc đánh ra những công thức phức tạp như vậy sẽ đơn giản và tiết kiệm thời gian hơn nhiều so với sử dụng các trình soạn thảo truyền thống.

Tuy nhiên, không thể phủ định khuyết điểm của LaTeX đó là: ta chỉ có thể biết được văn bản trông như thế nào sau khi hoàn tất quá trình biên dịch tập tin đầu vào. Người dùng có xu hướng muốn nhìn thấy kết quả tức thì khi họ soạn thảo để dễ dàng kiểm soát cả nội dung lẫn định dạng. Các trình soạn thảo LaTeX và máy tính ngày nay đã phần nào giải quyết được vấn đề trên, vì việc biên dịch và xử lý tập tin LaTeX giờ đã diễn ra gần như tức thì, chỉ với một nút bấm hay phím tắt, người soạn thảo có thể thấy được ngay văn bản đã định dạng của mình. Các trình soạn thảo nền web thậm chí còn cung cấp giao diện soạn thảo song song, giúp ta thấy văn bản kết quả cùng lúc với câu lệnh đánh ra.

2.2.2 Soạn thảo văn bản

Phần hướng dẫn dưới đây được tham khảo từ [22]. Người dùng có thể vào trang hướng dẫn của ShareLaTeX³, các diễn đàn như TeX Stack Exchange⁴ hoặc tài liệu [1] và [4] để tham khảo thêm nhiều tính năng và câu lệnh cũng như đặt câu hỏi về TeX và LaTeX.

³<https://www.sharelatex.com/learn>

⁴<https://tex.stackexchange.com/>

Cấu trúc cơ bản của một tập tin \LaTeX

Điều đầu tiên ta cần làm là tạo một tập tin `.tex`, sử dụng bất kì trình soạn thảo nào (ưu tiên sử dụng các trình soạn thảo \LaTeX vì chúng sẽ hỗ trợ gợi ý, tự động điền câu lệnh, highlight từ khoá và có các công cụ biên dịch). Ta xét ví dụ cơ bản sau:

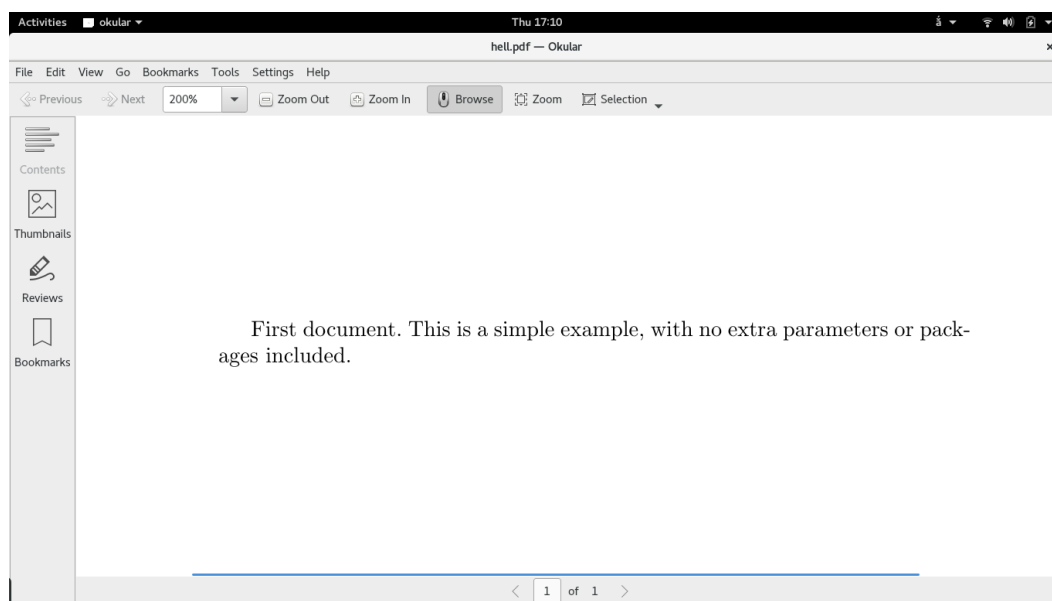
```
\documentclass{article}
```

```
\begin{document}
```

```
First document. This is a simple example, with no  
extra parameters or packages included.
```

```
\end{document}
```

Và được kết quả như hình 2.16.



Hình 2.16: Ví dụ một văn bản \LaTeX đơn giản

Ta có thể thấy \LaTeX tự động canh lề và thụt đầu dòng cho đoạn văn. Khi viết tập tin đầu vào, điều đầu tiên ta cần làm là khai báo lớp của văn bản với câu lệnh `\documentclass[option]{class}`, trong đó, *option* đóng vai trò như thông số (hoặc *tùy chọn, tính năng*) của lớp có thể được bỏ trống, mỗi lớp sẽ có các *option* khác nhau, và *class* là tên của lớp mà ta muốn sử dụng. Thông thường, tên lớp sẽ ứng với loại văn bản mà người dùng muốn soạn thảo như `article` (bài báo) hoặc `book` (sách), `report` (báo cáo),... Mỗi lớp sẽ có thêm các câu lệnh hoặc môi trường riêng (ngoài các câu lệnh chung cơ bản) quyết định định dạng và bố cục tổng thể của văn bản.

Tiếp đến là nội dung, nội dung bình thường của văn bản sẽ được viết giữa hai câu lệnh `\begin{document}` và `\end{document}`, bất kì câu lệnh hay nội dung nào được đánh vào giữa hai câu lệnh này đều được xem là “thân” (body) của văn bản và sẽ được hiển thị trong kết quả.

Tiếp theo, ta quay lại ví dụ đầu tiên ở 2.2.1 với một số thay đổi:

```
\documentclass[12pt, a4paper]{article}
\usepackage[utf8]{inputenc}
\usepackage[vietnamese]{babel}
\title{Tiêu đề của văn bản này}
\author{Nguyễn Văn A}
\date{Tháng 9 2015}
\begin{document}
  \maketitle
  Hello World!
\end{document}
```

Trong một tập tin LaTeX, phần trước câu lệnh `\begin{document}` được gọi là *tiền tố* (*preamble*). Ở phần tiền tố này, ta tiến hành các khai báo lớp, gói và sử dụng các câu lệnh với mục đích cung cấp thông số dưới dạng *từ khoá-giá trị* (*key-value*) hoặc nội dung (nếu có) cho gói và lớp.

```
\documentclass[12pt, a4paper]{article}
```

Ở đây lớp `article` đã được khai báo thêm hai tùy chọn, đó là **a4paper** và **12pt**. Các tùy chọn này phải được cách nhau bởi dấu phẩy (.). Trong ví dụ này, **12pt** là cỡ chữ, còn **a4paper** là khổ giấy sẽ được in ra trong văn bản đầu ra (output), ở đây là khổ giấy A4. Lớp `article` hỗ trợ các cỡ chữ **9pt**, **10pt**, **11pt** và **12 pt**, nếu để trống, mặc định sẽ là **10pt**. Ngoài khổ giấy A4, `article` còn hỗ trợ nhiều khổ giấy khác như **letterpage**, mọi chi tiết về các tùy chọn và thông tin của lớp này và tất cả các lớp cơ bản của LaTeX đều nằm trong tài liệu hướng dẫn [7].

```
\usepackage[utf8]{inputenc}
\usepackage[vietnamese]{babel}
```

Hai câu lệnh `\usepackage[option]{package}` cũng tương tự như `\documentclass` với hai đầu vào *option*, là các tùy chọn thêm không bắt buộc, và *package* là tên gói ta

muốn sử dụng. Không như `\documentclass` vốn là câu lệnh bắt buộc (và phải luôn được khai báo đầu tiên), ta không nhất thiết phải khai báo gói, tuy nhiên, khai báo gói vẫn rất cần thiết vì sẽ có trường hợp (nếu không muốn nói là thường xuyên) ta cần các gói bên ngoài bổ sung thêm nhiều chức năng cho LaTeX để hoàn thành các văn bản phức tạp.

Ví dụ trên sử dụng gói `inputenc`, đây là gói chứa các bộ mã hoá (encoding) các kí tự văn bản, tùy chọn `utf8` báo cho gói biết dùng chuẩn mã hoá UTF-8, ta thường không lường trước được trong văn bản sẽ xuất hiện kí tự đặc biệt gì hay không, cho nên gói này thường xuyên được sử dụng trong các văn bản. Gói tiếp theo là gói hỗ trợ đa ngôn ngữ `babel`. Khi văn bản có xuất hiện ngôn ngữ không phải tiếng anh, `babel` sẽ kết hợp với `inputenc` để xử lý ngôn ngữ đó, tùy chọn của gói `babel` phần nhiều là tên của các ngôn ngữ được định nghĩa trong tập tin của gói (như `vietnamese` hay `english`).

Mọi chi tiết của hai gói trên người dùng có thể tham khảo trong tài liệu hướng dẫn của từng gói `inputenc` [9], `babel` [15].

```
\title{Tiêu đề của văn bản này}
\author{Nguyễn Văn A}
\date{Tháng 9 2015}
```

Ba câu lệnh này cung cấp nội dung cho lớp `article` định dạng, chính vì thế chúng chỉ nên được đặt ở tiền tố, dựa trên định nghĩa của từng câu lệnh này, trong tập tin định nghĩa của lớp, mà LaTeX sẽ tiến hành định dạng mẫu nội dung được “đánh dấu”. Đúng như tên gọi, `\title{text}` dùng để cung cấp cho lớp tiêu đề văn bản, `\author{text}` là tác giả và `\date{text}` là thời gian văn bản được viết. Ba khai báo trên sẽ không xuất hiện trong văn bản đầu ra cho tới khi người dùng nhập lệnh `\maketitle` trong `\begin{document}`. Khi bắt gặp câu lệnh `\maketitle`, LaTeX sẽ định dạng các thông tin được cung cấp ở tiền tố và in ra kết quả.

Các định dạng cơ bản

Ta sẽ điểm qua ba câu lệnh định dạng cơ bản trong LaTeX:

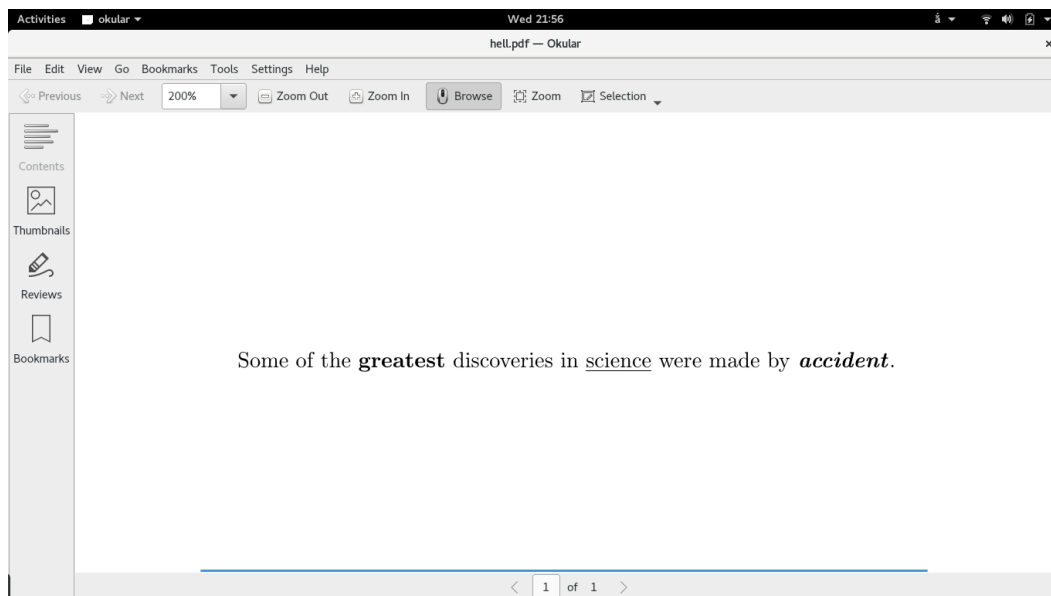
- **In đậm (Bold)**: Nhập đoạn cần in đậm vào câu lệnh `\textbf{text}` (viết tắt của text bold font).

- *In nghiêng (Italics)*: Ta sử dụng câu lệnh `\textit{text}` để đánh dấu đoạn văn bản cần in nghiêng.
- Gạch dưới (Underline): Câu lệnh `\underline{text}` giúp đánh dấu đoạn văn bản cần gạch dưới.

Ví dụ một đoạn sử dụng các lệnh trên như sau:

```
\begin{document}
Some of the \textbf{greatest}
discoveries in \underline{science}
were made by \textbf{\textit{accident}}.
\end{document}
```

Kết quả cho ra như hình 2.17:



Hình 2.17: Văn bản đầu ra được định dạng bằng các lệnh cơ bản

Ngoài ba câu lệnh trên, ta còn có `\emph{text}`, câu lệnh này sẽ định dạng đầu vào của nó phụ thuộc vào tình trạng định dạng của đoạn văn chứa nó, với đoạn văn có định dạng bình thường, `\emph` sẽ định dạng in nghiêng, dưới đây là ví dụ sử dụng câu lệnh ở các trường hợp khác nhau:

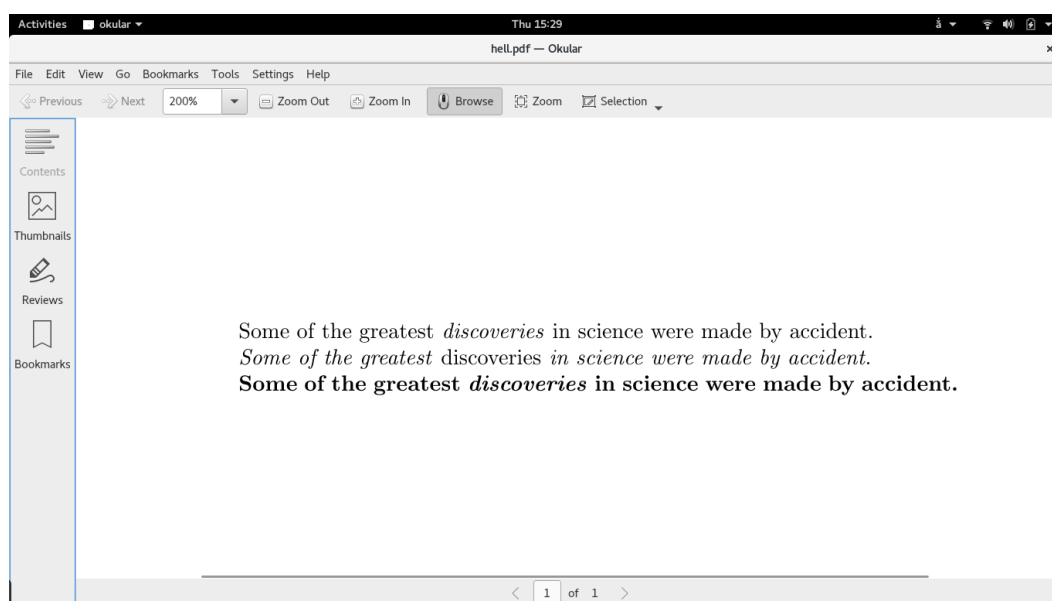
```

\begin{document}
Some of the greatest \emph{discoveries}
in science
were made by accident.

\textit{Some of the greatest \emph{discoveries}
in science
were made by accident.}

\textbf{Some of the greatest \emph{discoveries}
in science
were made by accident.}
\end{document}

```



Hình 2.18: Ví dụ cho câu lệnh `\emph`

Ở ví dụ hình 2.17, ta thấy mặc dù nội dung trong tập tin đầu vào có xuống dòng (linebreak) ở nhiều nơi, nhưng văn bản kết quả lại chỉ là một dòng duy nhất. Thông thường, LaTeX sẽ không xử lý “xuống dòng” trong văn bản nội dung, nếu ta cần phải xuống dòng ở một đoạn cụ thể nào đó, ta phải nhấn ENTER hai lần (xuống dòng hai lần) tại đoạn đó như ví dụ ở hình 2.18 hoặc sử dụng `\\` hay `\par` tại nơi ta muốn xuống dòng như ví dụ sau ở hình 2.19.

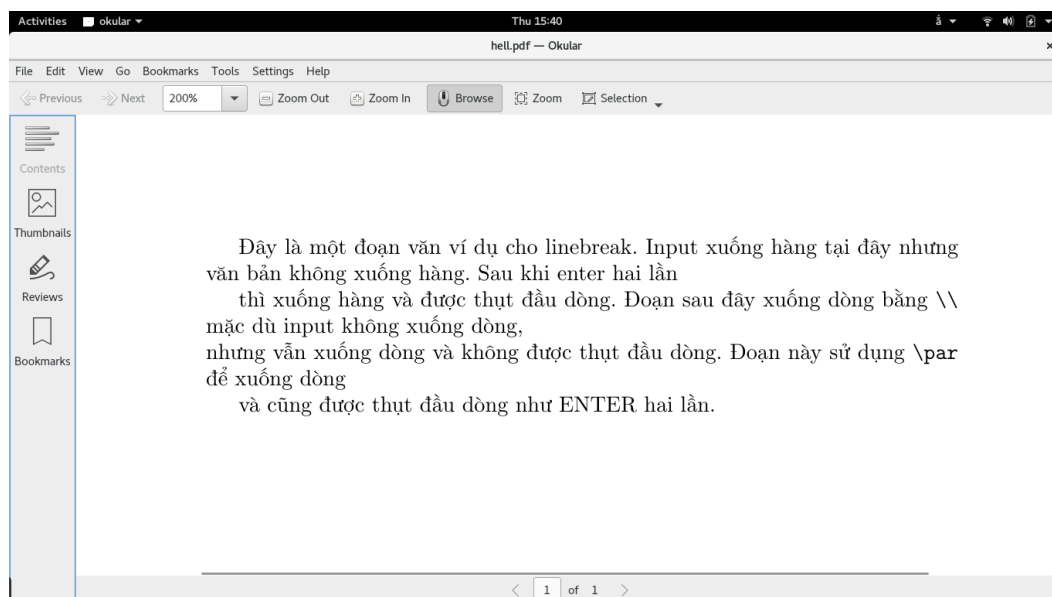
```
\begin{document}
```

Đây là một đoạn văn ví dụ cho linebreak. Input xuống hàng tại đây nhưng văn bản không xuống hàng. Sau khi enter hai lần

thì xuống hàng và được thụt đầu dòng. Đoạn sau đây xuống dòng bằng `\verb=\\` mặc dù input không xuống dòng, `\\` nhưng vẫn xuống dòng và không được thụt đầu dòng.

Đoạn này sử dụng `\verb=\par=` để xuống dòng `\par` và cũng được thụt đầu dòng như ENTER hai lần.

```
\end{document}
```



Hình 2.19: Ví dụ xuống dòng

Ta thấy văn bản đầu ra chỉ xuống dòng ở những nơi có ENTER hai lần, dấu `\\` và câu lệnh `\par`. Những nơi có ENTER chỉ một lần, LaTeX không cho xuống dòng và tiếp tục in thẳng cho tới khi hết chiều dài cho phép của một dòng. ENTER hai lần tương đương với câu lệnh `\par`, ngoài xuống dòng còn tiến hành canh chỉnh cho đoạn văn tiếp theo, dấu `\\` chỉ đơn thuần là cho xuống dòng. Ta thấy trong ví dụ hình 2.19, sau ENTER hai lần, dòng đầu của đoạn cũng thụt vào như `\par`.

Mặc định, văn bản LaTeX luôn ở chế độ canh đều (justify), nhưng nếu chúng ta muốn canh trái, phải hoặc giữa, LaTeX cũng cung cấp các câu lệnh và môi trường cho việc đó như ví dụ sau ở hình 2.20:


```
\begin{center}
```

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information.

```
\end{center}
```

```
\begin{flushleft}
```

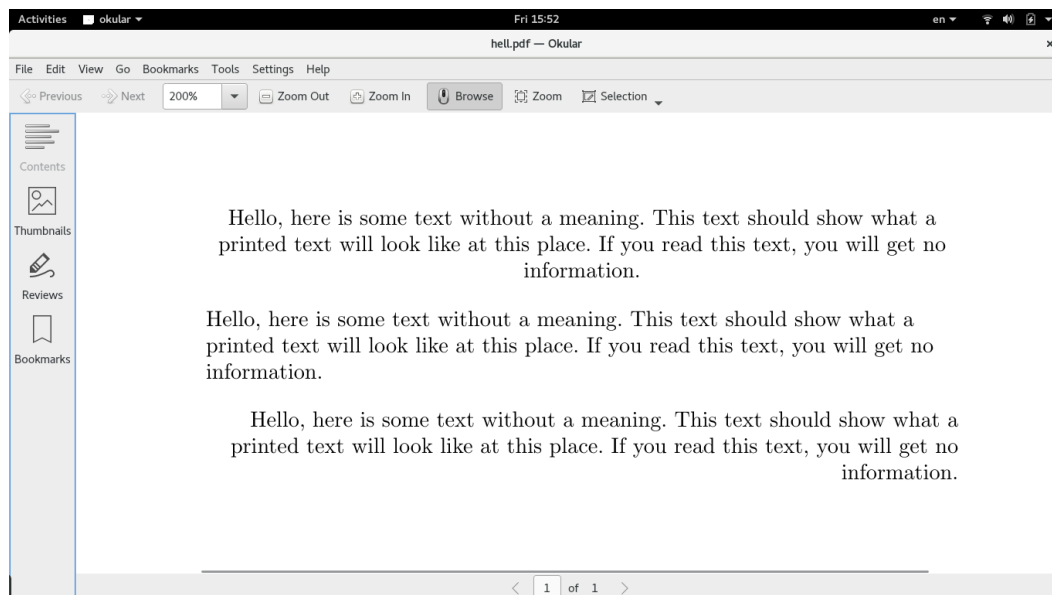
Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information.

```
\end{flushleft}
```

```
\begin{flushright}
```

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information.

```
\end{flushright}
```



Hình 2.20: Ví dụ canh vị trí

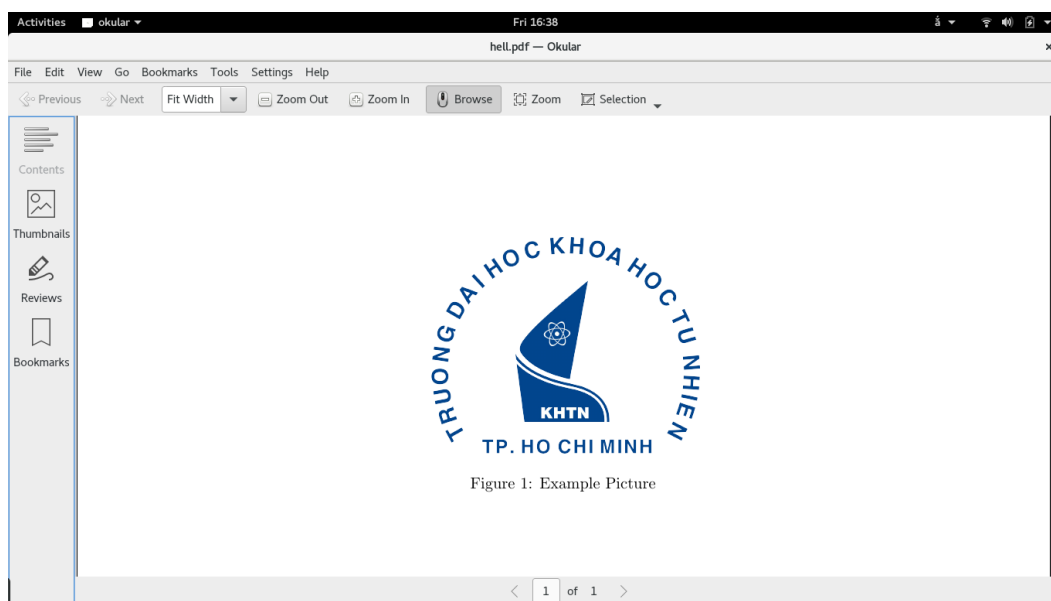
Như có thể thấy, ta chỉ cần đặt đoạn văn bản cần căn chỉnh vào giữa hai câu lệnh `\begin{env}...``\end{env}`, trong đó *env* là tên của môi trường (environment) canh chỉnh: ***center***, ***flushleft*** hoặc ***flushright***.

Thêm hình ảnh và tạo bảng

Dưới đây là đoạn input ví dụ cho việc thêm hình ảnh.

```
\documentclass{article}
\usepackage{graphicx}
\graphicspath{ {image/} }

\begin{document}
\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{logo}
\caption{Example Picture}
\end{figure}
\end{document}
```



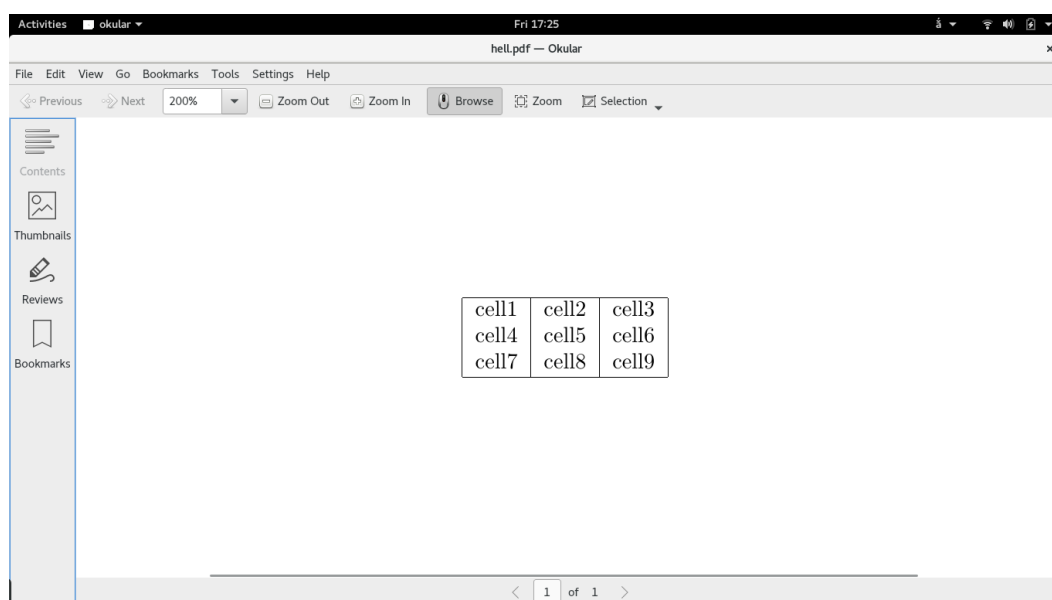
Hình 2.21: Ví dụ thêm hình trong văn bản

LaTeX cần phải tải (load) gói `graphicx` để thêm hình ảnh, đây là một trong số các gói cơ bản nên sẽ kèm theo trong gói phân phối của LaTeX. Để sử dụng gói này, ta chỉ việc khai báo `\usepackage{graphicx}`, tiếp theo là khai báo đường dẫn chứa tập tin hình bằng câu lệnh `\graphicspath{...}` trong phần tiền tố. Tại vị trí cần thêm hình, ta sử dụng môi trường **figure** với cú pháp `\begin{figure}[placement] ... \end{figure}`, trong đó, *placement* là vị trí đặt hình, ví dụ ở hình 2.21 sử dụng

h, tức đặt hình tại vị trí đặt câu lệnh (so với các đối tượng khác trong tập tin `.tex`). `\includegraphics[option]{name}` chính là câu lệnh thêm hình ảnh, với *option* ta có thể tùy chỉnh thông số (không bắt buộc) cho hình dưới dạng từ khoá-giá trị, ví dụ `width=0.9\textwidth`, tức chiều rộng hình bằng 0.9 chiều dài tối đa của 1 dòng trong văn bản, và *name* là tên hình ảnh ta muốn thêm. Câu lệnh `\caption{}` cho phép ta đặt chú thích cho hình, câu lệnh sẽ tự động đánh số và nhãn (label) tùy theo ngôn ngữ hiện hành, ở ví dụ `\caption` tự động điền label “Figure 1” trước chú thích hình.

Tiếp tục xét ví dụ tạo bảng trong LaTeX.

```
\begin{center}
\begin{tabular}{|c|c|c|}
\hline
cell1 & cell2 & cell3 \\
cell4 & cell5 & cell6 \\
cell7 & cell8 & cell9 \\
\hline
\end{tabular}
\end{center}
```



Hình 2.22: Ví dụ tạo bảng

Ta sử dụng môi trường ***tabular*** để tạo bảng với cú pháp `\begin{tabular}{cols}... \end{tabular}`, trong đó, *cols* là nơi ta khai báo số cột, canh chỉnh (trái, phải, giữa)

và đường phân cách. Trong ví dụ ở hình 2.22, `{ |c|c|c| }` ngụ ý bảng có 3 cột, nội dung của cột được canh giữa (center), ngoài c ta còn có các khai báo khác như l và r ứng với trái (left) và phải (right), kí hiệu | ám chỉ giữa mỗi cột sẽ có một đường gạch đơn dọc phân cách. Trong môi trường **tabular**, ta tiến hành soạn nội dung bảng với & đánh dấu sang cột mới và \\ đánh dấu sang dòng mới. \hline là câu lệnh báo LaTeX thêm đường kẻ ngang (ở vị trí đặt câu lệnh) cho bảng.

Để biết thêm nhiều câu lệnh liên quan tới việc thêm hình ảnh và tạo bảng, người dùng có thể tham khảo thêm tại trang web hướng dẫn của [ShareLaTeX](#) hoặc [1] và [4].

Chương và đề mục

Mỗi loại văn bản sẽ có cấu trúc sắp xếp nội dung khác nhau, tuy nhiên cách sắp xếp cơ bản nhất vẫn là chương (chapter) và đề mục (section).

```
\chapter{First Chapter}
```

```
\section{Introduction}
```

```
This is the first section.
```

```
Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Etiam lobortis facilisis sem. Nullam nec mi et
neque pharetra sollicitudin. Praesent imperdiet mi nec ante.
Donec ullamcorper, felis non sodales...
```

```
\section{Second Section}
```

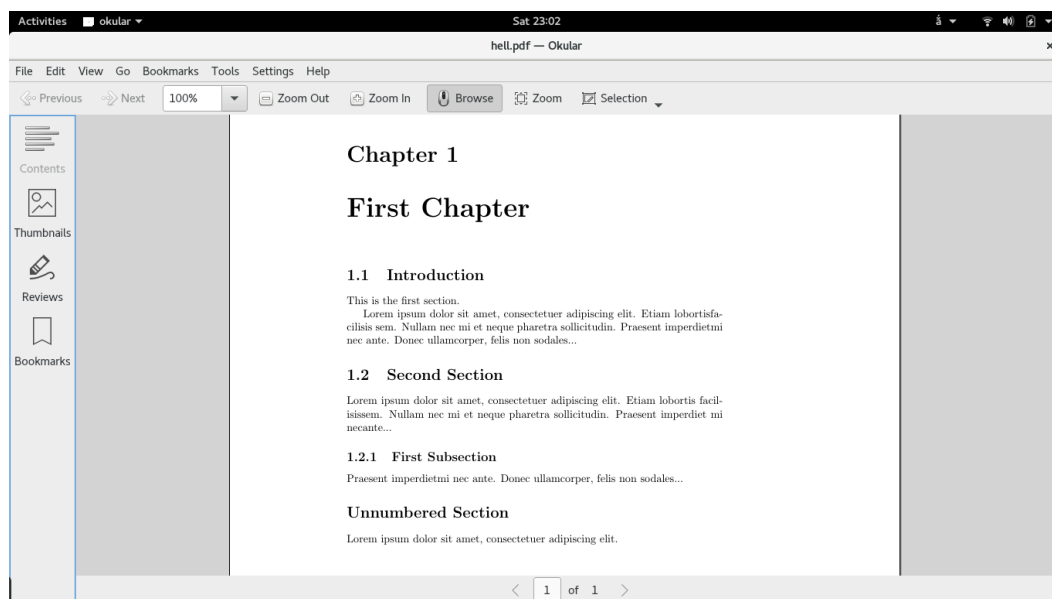
```
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra
sollicitudin. Praesent imperdiet mi nec ante...
```

```
\subsection{First Subsection}
```

```
Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales...
```

```
\section*{Unnumbered Section}
```

```
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
```



Hình 2.23: Đánh dấu chương và đề mục

Câu lệnh `\chapter{title}` và `\section{title}` đánh dấu bắt đầu một chương và mục, với *title* là tiêu đề của chương, mục đó. LaTeX sẽ tự động tiến hành định dạng cho tiêu đề và đánh số. Nếu không muốn đánh số ta có thể thêm dấu `*` vào câu lệnh như ví dụ hình 2.23 với câu lệnh `\section*`. Ngoài `\chapter` và `\section`, LaTeX còn có các câu lệnh chia đề mục khác ứng với thứ bậc như sau:

Thứ bậc	Câu lệnh
-1	<code>\part{}</code>
0	<code>\chapter{}</code>
1	<code>\section{}</code>
2	<code>\subsection{}</code>
3	<code>\subsubsection{}</code>
4	<code>\paragraph{}</code>
5	<code>\subparagraph{}</code>

Bảng 2.1: Các câu lệnh chương mục và thứ bậc

Chú ý: `\part` và `\chapter` chỉ có trong lớp văn bản `report` và `book`.

CHƯƠNG 3

Thiết kế định dạng văn bản riêng trong L^AT_EX

3.1 Tổng quan về gói và lớp

Như đã nói ở phần 1.3, người dùng hoàn toàn có thể tạo cho mình một lớp hoặc gói nhằm bổ sung chức năng, môi trường mới cho văn bản, tự mình quy định bố cục, định dạng tùy theo mục đích của văn bản mình muốn soạn thảo, hoặc chỉ đơn thuần tạo một quy chuẩn, mẫu có sẵn (form, template) cho những văn bản trình trọng, học thuật hay báo cáo khoa học,... Các gói và lớp được tác giả chia sẻ cho cộng đồng (La)TeX, sẽ được bảo hộ bằng giấy phép LPPL¹ và được lưu trữ trong Comprehensive T_EX Archive Network (CTAN), là tập hợp các trang web dùng để lưu trữ các gói, lớp của TeX và LaTeX, được xây dựng vào năm 1992. CTAN có các bản lưu trữ dự phòng (mirror) chính thức ở một số nước trên thế giới², danh sách các nguồn tải ở hình 2.5 là một trong số đó. Do được kết nối với các gói phân phối LaTeX, nên người dùng có thể tải các gói bổ sung trên CTAN thông qua chúng, giúp tác giả của gói, lớp có thể dễ dàng phân phối, bảo trì và cập nhật sản phẩm của mình.

Người dùng có thể dễ dàng tạo gói và lớp bằng bất cứ trình soạn thảo nào và chỉ cần đặt tên mở rộng của tập tin là .cls (lớp) hoặc .sty (gói), tuy nhiên, ta vẫn nên ưu tiên sử dụng trình soạn thảo hỗ trợ LaTeX với lý do tương tự như khi soạn thảo thông thường (được highlight từ khoá và gợi ý câu lệnh). Việc tiếp theo là tùy theo mục đích mà ta chọn tạo lớp hay gói. Thông thường, nếu muốn xây dựng, thiết kế bố cục, cấu trúc logic của một loại văn bản tùy ý nào đó, ta sẽ tạo lớp, mặt khác, nếu muốn xây dựng các câu lệnh, môi trường và tính năng mới để sử dụng được trong bất kì loại văn bản nào, ta xây dựng gói.

¹<http://www.latex-project.org/lppl.txt>

²<https://www.ctan.org/mirrors/>

3.2 Giới thiệu lớp `vlththesis`

Dựa vào giải thích trên và mục tiêu của đề tài này là thiết kế luận văn và bài báo cáo, nên sản phẩm cuối cùng của đề tài sẽ là một tập tin lớp LaTeX (LaTeX class file), với các câu lệnh, canh chỉnh theo đúng quy định về bố cục luận văn, bài báo cáo của trường và bộ môn. Lớp sản phẩm dựa trên lớp `book` của LaTeX, mọi chi tiết về lớp `book` xin hãy tham khảo tài liệu thông tin các lớp tiêu chuẩn của LaTeX [7]. Ý tưởng một số câu lệnh trong đề tài này dựa trên lớp `gsemthesis`, là mẫu (template) đề tài tiến sĩ của trường đại học Geneva, Thụy Sĩ, thông tin chi tiết về lớp này được nêu trong tài liệu hướng dẫn [10].

3.3 Cấu trúc của `vlththesis`

Phần này chủ yếu tập trung vào cấu trúc bên trong của `vlththesis` với các câu lệnh, khai báo, thông số cũng như các gói kèm theo, cùng các thiết lập cho chúng.

3.3.1 Các khai báo ban đầu

```
1 %Class for thesis writing
2
3 \NeedsTeXFormat{LaTeX2e}[1999/12/01]
4 \ProvidesClass{vlththesis}
```

Câu lệnh `\NeedsTeXFormat{LaTeX2e}[1999/12/01]`, trong đó `LaTeX2e` là bắt buộc [4], `[1999/12/01]` là phiên bản tối thiểu của LaTeX 2 ϵ mà người viết tập tin yêu cầu máy người dùng đáp ứng, hệ thống sẽ báo lỗi nếu người dùng sử dụng bản LaTeX cũ hơn phiên bản được nêu, do các gói phân phối tự động tải về bản LaTeX mới nhất, người dùng không cần quá quan tâm đến thông số này.

Câu `\ProvidesClass{vlththesis}` dùng để khai báo tên của lớp văn bản đang tạo và phải trùng với tên của tập tin `.cls` được lưu, ở đây, tên khai báo là “`vlththesis`” và tên của tập tin được lưu là `vlththesis.cls`.

```
5 \LoadClass[a4paper,oneside,12pt]{book}
```

Là câu lệnh dùng để tải lớp, mà sản phẩm ta xây dựng, sẽ dựa trên, như đã nói ở 3.2, lớp này dựa trên `book` của LaTeX, kèm theo các tùy chọn đó là cỡ giấy A4

(**a4paper**), in và định dạng một mặt giấy (**oneside**) và cỡ chữ **12pt** theo đúng chuẩn chung của một báo cáo khoá luận.

```
7 \usepackage[utf8]{inputenc}
8 \RequirePackage{etoolbox}
```

Lớp sử dụng gói `inputenc`, tùy chọn **utf8**, yêu cầu sử dụng bộ mã hoá (encoding) UTF-8 cho các kí tự trong văn bản LaTeX như đã giải thích ở 2.2.2.

`\RequirePackage{etoolbox}` dùng để nạp gói `etoolbox` chứa các câu lệnh hỗ trợ viết các tập tin lớp, gói cho LaTeX [16].

```
10 \DeclareOption{a5paper}{\PassOptionsToClass{\CurrentOption}{
    book}}
11 \DeclareOption{twoside}{\OptionNotUsed}
12 \newtoggle{viet}
13 \DeclareOption{vietnamese}{\toggletrue{viet}}
14 \newtoggle{noacr}
15 \DeclareOption{noacr}{\toggletrue{noacr}}
16 \newtoggle{print}
17 \DeclareOption{print}{\toggletrue{print}}
18 \DeclareOption*{\ClassWarning{vlththesis}{Unknown option '\
    CurrentOption'}}
19 \ProcessOptions\relax
```

Các câu lệnh `\DeclareOption{option name}{code}` trên dùng để khai báo các tùy chọn có trong lớp này mà người dùng có thể khai báo khi gọi lớp bằng câu lệnh `\documentclass[option]{class}`, trong đó, *code* là định nghĩa của tùy chọn, LaTeX sẽ tiến hành các câu lệnh trong phần này nếu người dùng khai báo tùy chọn *option name*. Câu lệnh ở dòng 19 (`\ProcessOptions\relax`) là câu lệnh bắt buộc sau khi kết thúc việc khai báo và định nghĩa các tùy chọn có trong lớp, nó có chức năng báo cho LaTeX phải xử lý các tùy chọn mà người dùng khai báo (nếu có) trước khi tiến hành các câu lệnh tiếp theo.

Các tùy chọn mà *vlththesis* có được định nghĩa thông qua các câu lệnh trên như sau:

```
10 \DeclareOption{a5paper}{\PassOptionsToClass{\CurrentOption}{
    book}}
11 \DeclareOption{twoside}{\OptionNotUsed}
```


Lớp cho phép sử dụng thêm tùy chọn **a5paper** của **book**, `\PassOptionsToClass{\CurrentOption}{book}` dùng để báo cho LaTeX truyền tùy chọn đó vào **book** nếu người dùng khai báo nó, tùy chọn này sẽ được viết đè (override) tùy chọn mặc định **a4paper** của lớp.

Câu lệnh `\DeclareOption{twoside}{\OptionNotUsed}` dùng để ra cảnh báo nếu người dùng sử dụng tùy chọn in hai mặt **twoside**, do chuẩn bài báo cáo chỉ yêu cầu in một mặt, người dùng có thể mở tập tin lớp này và thay đổi câu lệnh, sử dụng `\PassOptionsToClass{\CurrentOption}{book}`, nếu muốn sử dụng tùy chọn **twoside**.

```
12 \newtoggle{viet}
13 \DeclareOption{vietnamese}{\toggletrue{viet}}
```

Lớp có định nghĩa tùy chọn mới: **vietnamese**, `\newtoggle` và `\toggletrue` là hai câu lệnh của **etoolbox**, `\newtoggle{viet}` định nghĩa một biến boolean mới tên “viet” và khi người dùng khai báo tùy chọn **vietnamese**, lớp sẽ thực hiện câu lệnh `\toggletrue{viet}` trong phần định nghĩa và gán giá trị **true** cho biến **viet**.

```
14 \newtoggle{noacr}
15 \DeclareOption{noacr}{\toggletrue{noacr}}
16 \newtoggle{print}
17 \DeclareOption{print}{\toggletrue{print}}
```

Các câu lệnh trên cũng thực hiện điều tương tự với dòng 12 và 13, đặt giá trị **true** cho các biến boolean mới mỗi khi người dùng khai báo tùy chọn, chức năng hai tùy chọn này sẽ dần được nhắc đến ở các phần dưới.

```
18 \DeclareOption*{\ClassWarning{vlththesis}{Unknown option '\CurrentOption'}}
```

Dùng để hiển thị cảnh báo nếu người dùng khai báo tùy chọn không được định nghĩa, bằng câu lệnh `\ClassWarning{vlththesis}{Unknown option '\CurrentOption'}`. `\DeclareOption*{code}`, khác với `\DeclareOption`, chỉ nhận đầu vào *code* và dùng để xử lý bất kì tùy chọn nào, do đó nó thường được dùng để xử lý các tùy chọn không được định nghĩa hoặc dùng chung với câu `\PassOptionsToClass{\CurrentOption}{class}` để kế thừa toàn bộ các tùy chọn của lớp mà sản phẩm hiện hành dựa trên. LaTeX sẽ tiến hành tìm kiếm định nghĩa của tùy chọn được khai báo trước khi đẩy tùy chọn vào `\DeclareOption*{code}`, nếu không tìm thấy định nghĩa của tùy chọn cũng như `\DeclareOption*`, LaTeX vẫn tiếp tục chạy tập tin đầu vào nhưng sẽ hiển thị cảnh báo.

3.3.2 Khai báo gói và thiết lập

Dưới đây là các đoạn lệnh dùng để khai báo và thiết lập thông số cho các gói được sử dụng trong lớp này. Khai báo lớp đồng nghĩa với việc người dùng đã khai báo các gói tích hợp, người dùng hoàn toàn có thể thay đổi các thiết lập sau tùy thích sử dụng câu lệnh tương tự, khi soạn thảo tập tin đầu vào.

```

21 \iftoggle{viet}{%
22     \usepackage[vietnamese=nohyphenation]{hyphsubst} %Avoid
        hyphenation warning
23     \usepackage[vietnamese]{babel}%
24     \usepackage[indentfirst}
25 }{%
26     \usepackage[english]{babel}%
27 }
```

Câu lệnh điều kiện `\iftoggle{boolean}{true condition}{false condition}` của `etoolbox`, sẽ tiến hành kiểm tra biến trong *boolean*, nếu biến mang giá trị `true`, câu lệnh trong phần *true condition* được thực hiện, và ngược lại, lệnh ở phần *false condition* sẽ được triển khai nếu biến là `false`. Như đã nói ở 3.3.1, `viet` mang giá trị `true` khi người dùng khai báo tùy chọn **vietnamese**, khi đó, câu lệnh thực hiện phần *true condition*, tức ba câu lệnh từ dòng 22 tới 24.

Ba câu lệnh ở điều kiện `true`, dùng để nạp gói `babel` tùy chọn **vietnamese** để hỗ trợ xử lý tiếng Việt, người dùng phải tải thêm ngôn ngữ tiếng Việt cho `babel` bằng cách sử dụng `yum` hoặc `apt-get` kèm theo `install texlive-babel-vietnamese`. Gói `hyphsubst`, dùng để ngăn LaTeX đưa cảnh báo về ngắt chữ khi xuống dòng (tập tin đầu vào vẫn được chạy bình thường khi có cảnh báo này, việc sử dụng gói `hyphsubst` chỉ đơn thuần nhằm hạn chế các cảnh báo không cần thiết). Gói `indentfirst` cho phép LaTeX thụt đầu dòng đoạn văn (paragraph) đầu tiên của một chương, đề mục,...

Khi `viet` có giá trị `false` (tức người dùng không khai báo tùy chọn **vietnamese**), câu lệnh ở phần *false condition* sẽ được thực hiện, đó là nạp gói `babel` với tùy chọn **english** để hỗ trợ trình bày tiếng Anh.

Phần khai báo tiếp theo sẽ là về các gói phụ trợ không kèm theo các thiết lập thông số đặc biệt:

```

28 \usepackage[left = 1.38 in, right = 0.79 in, bottom = 1.15 in,
    top = 1.15 in]{geometry}
29 \usepackage[final]{graphicx}
30 \usepackage{multirow} %multirow table
31 \usepackage{wrapfig} %Wrapping fig and table with text
32 \usepackage{array} %extra options for table spec argument
33 \usepackage{float}
34 \usepackage{amsmath,amsfonts,amssymb}
35 \usepackage[nottoc]{tocbibind} %Add List of Figures, List of
    Tables,... to the Table of Content
36 \usepackage{notoccite} %suppress citation from appearing in ToC
37 \usepackage{path} %for file names, directory path, email...
    typesetting
38 \usepackage[table,figure]{totalcount}

```

Gói `geometry` dùng để canh chỉnh lề cho trang giấy và là một trong những gói cơ bản của LaTeX, có sẵn khi tải gói phân phối, tùy chọn của gói này thường là các thông số, kích thước cho lề của mặt giấy. Lớp sử dụng `geometry` để khai báo canh lề cho trang giấy với các thông số: left (lề trái) 1.38 inch, right (lề phải) 0.79 inch, top (lề đầu bao gồm header) 1.15 inch và bottom (lề cuối bao gồm footer) 1.15 inch.

Gói `graphicx` như đã nhắc tới ở phần 2.2.2, dùng để thêm hình ảnh vào văn bản và là một trong những gói cơ bản, có sẵn. Gói `multirow` cho phép tạo bảng gồm (merge) nhiều cột, dòng sử dụng trong môi trường ***tabular*** ở phần 2.2.2, chi tiết về cách gồm nhiều cột dòng người dùng có thể xem hướng dẫn, ví dụ ở ShareLateX [22] hoặc tài liệu của gói này [13].

Gói `wrapfig` cung cấp thêm môi trường ***wrapfigure*** và ***wraptable*** dùng để tạo hiệu ứng chữ bao lấy hình ảnh và bảng (giống với hiệu ứng Square của trình soạn thảo Microsoft Word), cách sử dụng môi trường này được cung cấp trong tài liệu [3] và các ví dụ sử dụng cũng được tìm thấy ở trang [22].

Gói `array` cung cấp thêm giá trị canh chỉnh mới cho ***tabular*** ngoài l, r và c (xem phần 2.2.2, phần hướng dẫn thêm hình và bảng để biết thêm chi tiết) đó là `mwidth` và `bwidth`, cho phép ta nhập đoạn văn dài vào bảng đồng thời xác định chiều rộng ô với `width`, chi tiết về gói này và môi trường ***tabular*** sẽ được tìm thấy trong tài liệu hướng dẫn [17].

Gói `float` cung cấp một biến thể của giá trị *placement* h, của môi trường ***figure***, đó là H. Biến thể này chặt chẽ hơn h, và dùng trong trường hợp h không đặt hình ở vị

trí mình mong muốn.

Các gói `amsmath`, `amsfonts`, `amssymb` cung cấp font chữ, câu lệnh, định dạng, canh chỉnh cho các công thức và kí hiệu toán học, cũng như văn bản cho các môi trường dùng để soạn thảo công thức (như ***equation*** hay ***align***), các gói này được chính **AMS** phát triển.

Gói `tocbibind` dùng để hiển thị “Danh sách hình ảnh” và “Danh sách bảng” lên mục lục, mặc định LaTeX sẽ không để các phụ chương không được “đánh số” (unnumbered) lên mục lục, `notoccite` dùng để hạn chế một số trường hợp các câu lệnh dẫn nguồn (`\cite`) bị liệt kê trên mục lục [2], gói `path` cung cấp câu lệnh `\path| |`, cho ta đánh dấu đường dẫn, câu lệnh, email, ... trong văn bản LaTeX, nội dung bên trong sẽ được định dạng bằng font chữ `monospace` và các kí tự đặc biệt trong đó sẽ được xem như kí tự bình thường (giống như khi ta “chú thích” trong lập trình). Gói `totalcount` dùng để kiểm tra và đếm số lượng bảng, hình trong văn bản, người dùng bình thường không cần phải quá quan tâm đến gói này.

Lưu ý: Các gói `multirow`, `wrapfig`, `tocbibind`, `notoccite`, `path` có thể không có sẵn trong gói phân phối, người dùng cần phải tải bổ sung nếu hệ thống báo thiếu, cách tải gói LaTeX bổ sung trong Linux được nêu ở phần 2.1.1.

Tiếp theo là các gói kèm theo các thiết lập thông số:

```
40 \usepackage[obeyspaces]{url} %typesetting url
41 \urlstyle{sf}
```

Tương tự như gói `path`, gói `url` cung cấp câu lệnh `\url| |` để ta đánh dấu địa chỉ URL của website, tùy chọn ***obeyspaces*** cho phép hiển thị khoảng cách (nếu có) trong địa chỉ (hiếm khi xảy ra), gói còn cung cấp câu lệnh `\urlstyle` cho phép ta đề ra các thiết lập định dạng cho URL như font, cỡ chữ, ... Ở đây, các URL được đánh dấu bằng `\url| |` sẽ có font chữ Sans Serif (sf). Người dùng có thể thay đổi thiết lập này, sử dụng câu lệnh tương tự, các thay đổi có thể được thực hiện bằng câu lệnh này được liệt kê trong tài liệu [6] của gói.

```
43 \usepackage{titlesec} %customize chapter and sections'... title
```

Tiếp theo là gói `titlesec` dùng để định dạng đề mục, đây là gói có sẵn trong gói phân phối, lớp này sử dụng hai câu lệnh chính của `titlesec` là `\titleformat` và `\titlespacing*`, lưu ý rằng `\titlespacing*` chỉ có thể sử dụng được khi có `\titleformat` trước nó [11]. Ta xét cú pháp hai câu lệnh đó:

```
\titleformat{command}[shape]{format}{label}{sep}{before-code}[after-
code]
```

Trong đó:

- `<command>` là câu lệnh đề mục mà ta muốn thay đổi, như `\part`, `\chapter`, `\section`,...
- `<shape>` là kiểu bố trí thành phần đề mục như `hang`, `block`, `display`,... chi tiết về các kiểu bố trí được nêu trong [11].
- `<format>` dùng để định dạng font, cỡ chữ, canh chỉnh cho nhãn và tiêu đề chương.
- `<label>` dùng để quy định cách trình bày nhãn của đề mục ví dụ “Chương 2” thành “Chương II.”.
- `<sep>` là khoảng cách giữa nhãn và tiêu đề của đề mục.
- `<before-code>` và `<after-code>` dùng để bổ sung những câu lệnh ta muốn LaTeX thực hiện trước và sau khi trình bày tiêu đề.

```
\titlespacing*{command}{left}{before-sep}{after-sep}[right-sep]
```

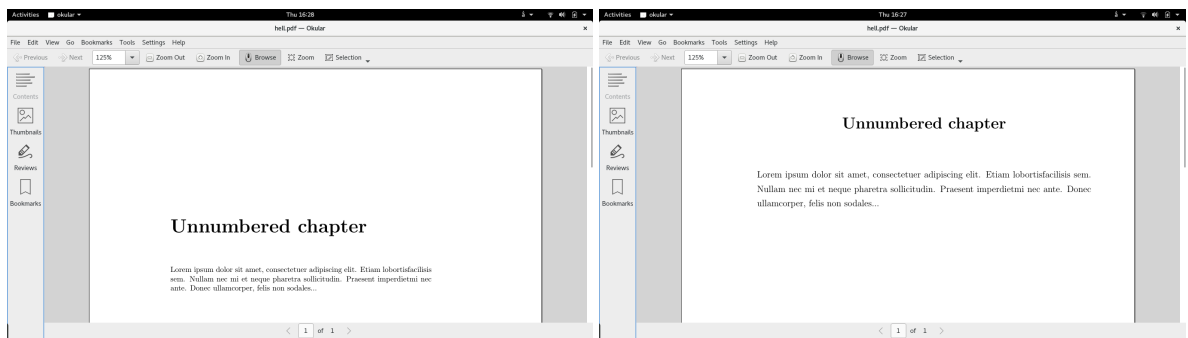
Trong đó:

- `<command>` tương tự như với `\titleformat`.
- `<left>` khoảng cách giữa lề trái và tiêu đề.
- `<before-sep>` là khoảng cách từ lề đầu của trang đến tiêu đề.
- `<after-sep>` là khoảng cách giữa tiêu đề và nội dung văn bản.
- `<right-sep>` khoảng cách giữa lề phải và tiêu đề.

Hai câu lệnh trên được sử dụng trong lớp này nhằm thay đổi bố cục mặc định của tiêu đề trước tiên là các đề mục phụ (không chia số chương như “Lời giới thiệu”, “Lời cảm ơn”,...) còn gọi là `unnumbered` hay `numberless chapter`.

```
44 \titleformat{name=\chapter,numberless}[display]
45 {\normalfont\filcenter\bfseries}{}{0pt}{\LARGE}
46
47 \titlespacing*{name=\chapter,numberless}{0pt}{-55pt}{40pt}
```

Câu `\titleformat` trên sử dụng thông số định dạng tương tự với định dạng mặc định của LaTeX dành cho đề mục không đánh số, chỉ có một số thay đổi đó là câu lệnh `\filcenter` để canh giữa và cỡ chữ từ `\Huge` sang `\LARGE`, nhỏ hơn hai bậc so với `\Huge`, để thu nhỏ tiêu đề so với định dạng mặc định. Do `\titlespacing*` cần phải đi kèm với `\titleformat` trước, nên giả sử nếu ta chỉ muốn thay đổi khoảng cách tiêu đề, ta cũng phải sử dụng `\titleformat` với các định dạng mô phỏng lại giống hoặc gần giống với định dạng mặc định, với `\titlespacing*`, lớp cũng thay đổi khoảng cách từ lề đầu đến tiêu đề, số âm ở giá trị cho thấy, đề mục lấn ra phần header, tuy nhiên, do tiêu đề ở trang đầu của một chương được coi là header, nên lớp tạm thời “mượn” khoảng đó để đặt đề mục. Kết quả có được khi đặt hai câu lệnh đó và sử dụng lớp này thay cho lớp `book` thông thường được so sánh ở hình 3.1:

(a) Sử dụng lớp `book`(b) Sử dụng lớp `vlththesis`**Hình 3.1:** So sánh đề mục của hai lớp văn bản

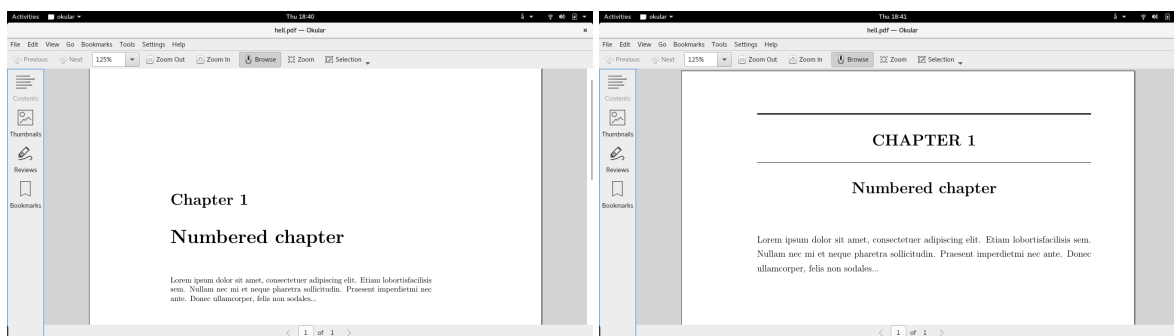
Tiếp theo là câu lệnh dành cho đề mục chương có đánh số (numbered chapter):

```

49 \titleformat{name=\chapter}[display]
50 {\normalfont\LARGE\filcenter\bfseries}
51 {\titlerule[1pt]}%
52 \vspace{1pt}%
53 \titlerule
54 \vspace{1pc}%
55 \LARGE\MakeUppercase{\chaptertitlename} \thechapter}
56 {1pc}
57 {\titlerule
58 \vspace{1pc}%
59 \LARGE}
60
61 \titlespacing*{name=\chapter}{0pt}{-40pt}{40pt}

```

Đề mục chương đánh số cũng có thông số `\titlespacing*` gần giống với không đánh số, sự khác nhau là do chương đánh số có kèm theo nhãn (chữ “Chương...” hay “Chapter...”), phần `\titleformat` dùng để canh giữa và bổ sung thêm các đường kẻ (ruler) và thay đổi cỡ chữ nhỏ hơn so với mặc định. Kết quả được trình bày ở hình 3.2:



(a) Sử dụng lớp book

(b) Sử dụng lớp vlththesis

Hình 3.2: So sánh đề mục (có đánh số) của hai lớp văn bản

Tiếp theo là gói `caption` và `subcaption`:

```

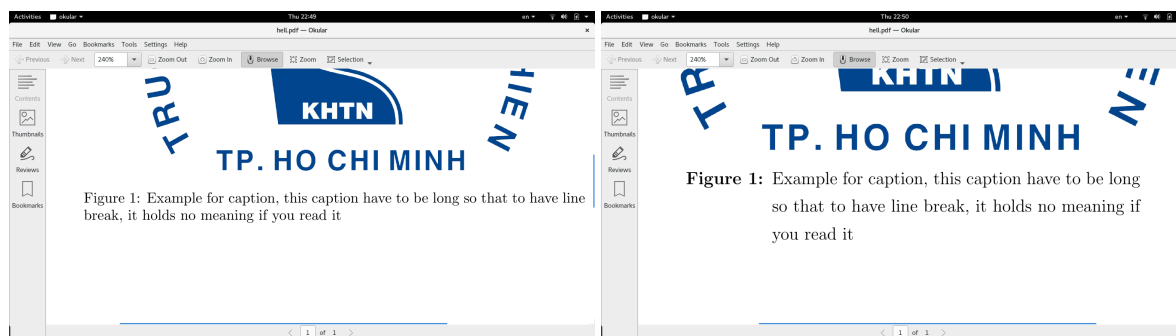
63 %packages for caption
64 \usepackage{caption}
65 \usepackage{subcaption}
66 \captionsetup{
67     labelfont=bf,
68     format=hang,
69     width=0.8\linewidth
70 }

```

Trong đó gói `caption` cho phép ta định dạng chú thích cho bảng và hình (caption) thông qua câu lệnh `\captionsetup` bằng cách sử dụng khai báo kiểu từ khoá-giá trị như trên từ font chữ (*font*, *labelfont*, *textfont*), định dạng (*format*), và độ rộng cho phép (*width*),... Để biết thêm các từ khoá cũng như giá trị tương ứng mà ta có thể dùng để định dạng caption, người dùng có thể tham khảo thêm trong tài liệu [5].

Gói `subcaption` là gói đi kèm với `caption` và cả hai đều có sẵn trong gói phân phối, `subcaption` cho phép ta chèn nhiều hình phụ với một caption chung, ví dụ cho dạng trình bày này chính là các hình 2.1, 3.1 và 3.2 ở trên, bên cạnh đó gói cho phép các hình phụ có chú thích riêng với môi trường **subfigure** do nó cung cấp, ngoài ra gói còn bổ sung thêm tùy chọn `sub` cho câu lệnh `\captionsetup` (`\captionsetup[sub]`) cho phép chú thích phụ có định dạng riêng.

Nhờ tích hợp câu lệnh trên mà lớp này có định dạng chú thích khác với book.



(a) Sử dụng lớp book

(b) Sử dụng lớp vlththesis

Hình 3.3: So sánh chú thích của hai lớp văn bản

Các dòng tiếp theo là khai báo và sử dụng gói `xcolor`:

```
72 \usepackage{xcolor} %define new color
73 \definecolor{cstblue}{HTML}{3333FF}
74 \definecolor{cstred}{HTML}{FF1A1A}
75 \definecolor{cstgreen}{HTML}{00B300}
76 \definecolor{maroon}{HTML}{800000}
77 \definecolor{lstblue}{HTML}{00008B}
78 \definecolor{lstgray}{HTML}{787878}
```

Ở đây tôi sử dụng câu lệnh `\definecolor` để định nghĩa một số màu sắc sẽ sử dụng trong lớp này. Gói `xcolor` cho phép ta tạo màu sử dụng các mã màu thông dụng như HTML, RGB, ... và cung cấp câu lệnh cho phép ta định màu cho một câu chữ hay đoạn văn bản, nếu người dùng quan tâm muốn biết thêm các chức năng câu lệnh có trong gói có thể tham khảo trong [12] hoặc trang web hướng dẫn ShareLaTeX [22].

Tiếp đến là các khai báo cho gói `hyperref`.

```
80 \usepackage[linktocpage=true]{hyperref}
81 \hypersetup{%
82     colorlinks=true,
83     breaklinks=true,
84     linkcolor=cstred,
85     filecolor=black,
86     urlcolor=cstblue,
87     citecolor=cstgreen%
88 }
```



```

89 \iftoggle{print}{\hypersetup{%
90     colorlinks=false,
91     breaklinks=true}}

```

Gói `hyperref`, dùng để tạo, thiết lập thuộc tính cho các siêu liên kết (hyperlink) trong văn bản (liên kết nội, url, ...), và là một gói cơ bản có sẵn, tùy chọn `linktocpage=true` để biến số trang trên mục lục thành liên kết dẫn đến trang đó (thay vì mặc định là cả tiêu đề trong mục lục là liên kết).

Tương tự như `caption`, `hyperref` cung cấp câu lệnh `\hypersetup` cho ta định màu sắc cho từng loại liên kết, tinh chỉnh hành vi, định dạng của chúng bằng từ khoá-giá trị. Trong lớp này, tôi sử dụng những màu sắc đã tạo bằng `\definecolor` cho các loại liên kết khác nhau như liên kết nội bộ (`linkcolor`), liên kết tập tin bên ngoài (`filecolor`), liên kết url (`urlcolor`) và liên kết dẫn nguồn (`citecolor`). `colorlinks=true` dùng để bật tính năng màu cho liên kết và phải `true` nếu người dùng muốn định màu cho các liên kết và `breaklinks=true` cho phép liên kết được xuống hàng như văn bản bình thường.

Câu `\iftoggle{print}` để kiểm tra người dùng có khai báo tùy chọn **print** của lớp hay không, nếu biến boolean `print` là `true` (người dùng có khai báo) tiến hành tắt tính năng tô màu cho liên kết (`colorlinks=false`), tuy nhiên, các liên kết đó vẫn được đánh dấu bằng các ô vuông theo màu mặc định của LaTeX (không phải màu do người dùng định), mục đích của tính năng này là do khi in văn bản ra, các màu link vẫn được giữ nguyên trên bản in nếu người dùng in màu, vì vậy cần phải bổ sung thêm tùy chọn này để tắt màu liên kết.

```

93 \usepackage{fancyhdr}
94 \pagestyle{fancy}
95 \fancyhf{}
96 \fancyhead[R]{\slshape\nouppercase{\rightmark}}
97 \fancyhead[L]{\slshape\nouppercase{\chaptertitlename\
    thechapter}}
98 \fancyfoot[C]{\thepage}
99 \renewcommand{\headrulewidth}{0.5pt}

```

Gói `fancyhdr` dùng để tạo và định dạng header và footer, với các câu lệnh `\fancyhead` và `\fancyfoot`, người dùng có thể quy định vị trí và nội dung hiển thị trên header và footer (như số trang, tên chương, ...). Với gói này, người dùng có thể đặt bố cục cho một trang nhất định hay cả văn bản (sử dụng lệnh `\pagestyle` hoặc `\thispagestyle`), định dạng, đặt tên bố cục header, footer cho riêng mình với câu

lệnh `\fancypagestyle`. Các kiểu bố cục (style) có sẵn thường được dùng là: `empty` (không header hay footer), `plain` (chỉ có số trang ở giữa footer) và `fancy` (do người dùng định ra sử dụng `\fancyhead` và `\fancyfoot`), ngoài ra còn có `myheadings`, chi tiết về các mẫu bố cục và các thao tác với gói này có thể được tìm thấy trong tài liệu [18].

Ở đoạn code trên, ta có `\pagestyle{fancy}` dùng để đặt kiểu `fancy` cho cả văn bản, kể đến là phần thiết kế bố cục, trong đó, `\fancyhf{}` để xoá các header, footer hiện hành, đây chỉ là bước reset trước khi định dạng, câu lệnh `\fancyhead[R]` dùng để đặt `\rightmark` (tức tiêu đề và nhãn mục của đơn vị chương hồi *thấp nhất* của trang nội dung hiện t, xem bảng 2.1 để biết thêm về thứ bậc chương) vào vị trí bên phải của header (R ứng với vị trí phải). Câu lệnh `\fancyhead[L]` để đặt `\chaptertitlename\thechapter` vào vị trí bên trái (L=Left), trong đó, `\chaptertitlename` chính là “Chương” hay “Chapter” (tùy theo ngôn ngữ hiện hành của `babel` mà chữ này có thể khác nhau) và `\thechapter` chính là số chương hiện hành của trang. Tiếp theo, `\fancyfoot[C]` dùng để đặt `\thepage`, tức số trang ở phần giữa (C=Center) cho footer, cuối cùng là `\renewcommand{\headrulewidth}` để tạo đường kẻ cho header.

Lưu ý: Nếu không nêu rõ các thiết lập trên là của style gì (bằng cách sử dụng câu lệnh `\fancypagestyle{tên style}{các thiết lập}`) gói sẽ mặc định hiểu đó là của kiểu `fancy`.

Sau khi tiến hành thực hiện các câu lệnh trên, kết quả có được chính là bố cục header và footer của cuốn báo cáo này. Ngoài bố cục này, lớp còn có thêm bố cục khác dùng cho phụ lục:

```

101 \fancypagestyle{supplement}{%
102     \fancyhf{}
103     \fancyhead[L]{\slshape\nouppercase{\leftmark}}
104     \fancyhead[R]{\thepage}
105     \renewcommand{\headrulewidth}{0.5pt}%
106 }
```

Tiếp theo là gói hỗ trợ nhập mã (code) vào văn bản LaTeX.

```

109 \usepackage{listings} %for source code input
```

Đây là gói cung cấp các câu lệnh dùng để định dạng từ khoá và hỗ trợ in code cho văn bản LaTeX, vốn được cộng đồng xem là một bảng nâng cấp của môi trường **`verbatim`** và câu lệnh `\verb` của LaTeX. Câu lệnh cung cấp môi trường **`lstlisting`** với các tùy chọn dạng từ khoá-giá trị cho phép in số dòng cho code theo nhiều cách tùy

thích, nhận dạng câu lệnh, từ khoá của ngôn ngữ lập trình (danh sách các ngôn ngữ được hỗ trợ có trong [8]) từ đó cho người dùng định màu sắc cho các loại từ khoá đó như trường hợp các loại liên kết trong `hyperref`, người dùng thậm chí còn có thể định danh ngôn ngữ không được hỗ trợ sẵn trong gói, như trường hợp sau đây là một đoạn lệnh có trong `vlththesis` dùng để định nghĩa ngôn ngữ JavaScript do cộng đồng LaTeX chia sẻ:

```

111 \lstdefinlanguage{JavaScript}{
112   keywords={typeof, new, true, false, catch, function, return,
            null, catch, switch, var, if, in, while, do, else, case,
            break},
113   ndkeywords={class, export, boolean, throw, implements, import
            , this},
114   ndkeywordstyle=\bfseries,
115   sensitive=false,
116   comment=[l]{//},
117   morecomment=[s]{/*}{*/},
118   morestring=[b]',
119   morestring=[b]"
120 }
```

Ta có thể thấy, ngôn ngữ được định nghĩa bằng các giá trị từ khoá-giá trị trong câu lệnh `\lstdefinlanguage`, ý nghĩa của các giá trị này bao gồm liệt kê từ khoá của ngôn ngữ lập trình (`keywords`, `ndkeywords`), ngôn ngữ có phân biệt chữ hoa và thường hay không (`sensitive`), định ra kí hiệu ghi chú (`comment`) của ngôn ngữ (`comment`, `morecomment`) và kí hiệu chuỗi (`morestring`).

Cũng như các gói `caption`, `hyperref`,... gói này cũng có `\lstset` giúp ta chỉnh sửa định dạng cho môi trường *lstlisting*, các định dạng sử dụng trong lớp này như sau:

```

123 \lstset{%
124   basicstyle=\ttfamily\small,
125   keywordstyle=\color{lstblue}\bfseries,
126   identifierstyle=,
127   frame=,
128   commentstyle=\color{lstgray},
129   stringstyle=\color{maroon},
130   numbers=left,
131   breakatwhitespace=false,
```

```

132 breaklines=true ,
133 captionpos=b ,
134 keepspaces=true ,
135 showspaces=false ,
136 showtabs=false ,
137 escapeinside=,
138 showstringspaces=false ,
139 inputencoding=utf8%
140 }

```

Các định dạng bao gồm kiểu, cỡ chữ hay màu sắc chung cho toàn bộ đoạn mã (`basicstyle`), cho các từ khoá của ngôn ngữ lập trình (`keywordstyle`), cho các ghi chú (`commentstyle`) và chuỗi (`basicstyle`), tùy chỉnh vị trí hoặc bật tắt đánh số dòng (`numbers`),... Một số từ khoá để trống nhằm gợi ý tính năng cho những ai có mong muốn tinh chỉnh định dạng của `lstlisting`, tất nhiên các định dạng này có thể được ghi đè (`override`) hoặc bổ sung bởi `\lstset` của người dùng trong tập tin đầu vào, ngoài ra ta còn có thể khai báo các từ khoá-giá trị này ngay khi khai báo môi trường bằng cú pháp `\begin{lstlisting}[key-value list]...\end{lstlisting}`, điều này thích hợp khi ta muốn nhiều định dạng khác nhau cho nhiều đoạn code khác nhau, ngoài các từ khoá-giá trị của `\lstset` sử dụng được trong *key-value list* trên, môi trường ***lstlisting*** cũng có các từ khoá-giá trị riêng. Tài liệu [8] cung cấp đầy đủ các câu lệnh và các giá trị định dạng mà gói cung cấp. Ta xét ví dụ một đoạn văn bản LaTeX sau, được tạo trên *vlththesis*, dùng để nhập một đoạn mã C:

```

\begin{lstlisting}[language=C,title=Code C example, frame=single]
#include <stdio.h>
int main()
{
    int firstNumber, secondNumber, sumOfTwoNumbers;

    printf("Enter two integers: ");

    // Two integers entered by user is stored using scanf() function
    scanf("%d %d", &firstNumber, &secondNumber);

    // sum of two numbers in stored in variable sumOfTwoNumbers
    sumOfTwoNumbers = firstNumber + secondNumber;

```

```
// Displays sum
printf("%d + %d = %d", firstNumber, secondNumber, sumOfTwoNumbers);

return 0;
}
\end{lstlisting}
```

Và kết quả ta được:

```
1 #include <stdio.h>
2 int main()
3 {
4     int firstNumber, secondNumber, sumOfTwoNumbers;
5
6     printf("Enter two integers: ");
7
8     // Two integers entered by user is stored using scanf()
function
9     scanf("%d %d", &firstNumber, &secondNumber);
10
11     // sum of two numbers is stored in variable sumOfTwoNumbers
12     sumOfTwoNumbers = firstNumber + secondNumber;
13
14     // Displays sum
15     printf("%d + %d = %d", firstNumber, secondNumber,
sumOfTwoNumbers);
16
17     return 0;
18 }
```

Code C example

Ta có thể thấy, ví dụ đã bổ sung thêm các tùy chỉnh [language=C,title=Code C example, frame=single] vào những định dạng có sẵn trong `\lstset`. Điều này cho thấy người dùng hoàn toàn có thể bổ sung, thay đổi các định dạng có sẵn trong lớp.

Các khai báo sau đây là dành cho gói `biblatex`.

```

142 \usepackage{csquotes}
143 \usepackage[backend=biber,
144             style=numeric,
145             sorting=ynt,
146             citestyle=numeric-comp
147           ]{biblatex}

```

Gói `csquotes` dùng để cung cấp các công cụ quản lý các dấu trích dẫn câu (quote), được sử dụng trong lớp này nhằm loại bỏ cảnh báo về dấu khi sử dụng `biblatex`.

Gói `biblatex` là gói dùng trong việc tạo, sắp xếp và in danh sách các tài liệu tham khảo, hỗ trợ chức năng và kiểu dẫn nguồn, tự động định dạng và trình bày các thông tin được cung cấp. Gói này không có sẵn và người dùng buộc phải tải về.

Để lập danh sách tài liệu tham khảo, người dùng phải tạo tập tin `.bib`, sau đó sử dụng câu lệnh `\addbibresource{<tên file>.bib}` để trỏ đường dẫn tới tập tin đó, sau đó là sử dụng `\printbibliography` tại vị trí ta muốn in danh sách (so với các đối tượng khác trong văn bản LaTeX). Trong tập tin `.bib`, ta sử dụng một dạng khai báo đặc biệt để cung cấp cho LaTeX thông tin về các tài liệu như ví dụ sau.

```

@book{latex-comp,
title={The LaTeX Companion},
author={Frank Mittelbach and Michel Goossens},
edition=2,
year=2004,
isbn={0-201-36299-6},
publisher={Addison-Wesley Professional},
pagetotal=1120,
}

```

Trong đó `@book` cho biết tài liệu này là sách, `latex-comp` là nhãn ta gán cho tài liệu này và sẽ được gọi ra sử dụng câu lệnh `\cite{nhãn}` tại đoạn văn ta muốn dẫn nguồn, `title` là nhan đề đầy đủ của tài liệu, `author` là (các) tác giả của tài liệu, `edition` là phiên bản mà người dùng tham khảo, `year` là năm xuất bản, `isbn` là số hiệu **ISBN** của tài liệu, `publisher` là nhà xuất bản và `pagetotal` là tổng số trang.

Trong các thông tin trên, chỉ có `author`, `title`, `year` là những thông số bắt buộc, các thông số còn lại người dùng có thể lược bớt tùy thích. Để biết thêm cách khai báo nhiều loại tài liệu khác nhau và các thông tin mà ta có thể khai báo cho loại

tài liệu nào đó, kèm theo các câu lệnh có thể dùng để thao tác với gói này, người dùng có thể tham khảo thêm trong tài liệu [14]. Lưu ý, `biblatex` chỉ in và liệt kê những tài liệu mà người dùng có dẫn nguồn *ít nhất một lần* trong văn bản bằng câu lệnh `\cite`.

Ta quay lại khai báo gói của lớp.

```

143 \usepackage[backend=biber,
144           style=numeric,
145           sorting=ynt,
146           citestyle=numeric-comp
147           ]{biblatex}

```

Các tùy chọn chủ yếu dùng để quy định chương trình backend (loại module dùng để chuyển dữ liệu từ mã nguồn `biblatex` sang mã LaTeX [14]), ở đây sử dụng backend `biber` vốn có sẵn trong các gói phân phối LaTeX, các từ khoá `style`, `citestyle` dùng để khai báo kiểu dẫn nguồn (sử dụng số hay chữ viết tắt) và `sorting` là quy định cách sắp xếp các tài liệu tham khảo, ở đây khai báo kiểu sắp xếp `ynt`, tức sắp xếp theo năm xuất bản (year), tên tác giả (name) và tiêu đề (title). Các kiểu dẫn nguồn và sắp xếp đều được liệt kê và giải thích rõ ràng trong [14].

```

149 \defbibheading{bibliography}[\refname]{%
150 \iftoggle{viet}{\renewcommand{#1}{Tài liệu tham khảo}}{~}
151 \chapter*{#1}
152 \markboth{#1}{#1}
153 \addcontentsline{toc}{chapter}{#1}}

```

Câu lệnh `\defbibheading` dùng để thiết lập các thuộc tính tiêu đề cho danh sách tài liệu tham khảo và lưu thiết lập đó sử dụng biệt hiệu, theo mặc định, `biblatex` sử dụng style `bibliography`, chính vì thế ta trực tiếp thay đổi style `bibliography` thay vì đặt ra tên cho style mới. Ở đây, ta sửa lại mặc định gốc của style `bibliography`, thay đổi macro mà style này sử dụng từ `\bibname` sang `\refname`, macro `\refname` lưu giá trị “Reference” cho tiêu đề của danh sách tài liệu tham khảo, ta thêm vào một câu lệnh điều kiện `\iftoggle{viet}` để nếu người dùng có khai báo tùy chọn `vietnamese` sẽ tiến hành thay đổi giá trị trong `\refname` thành “Tài liệu tham khảo”, ngoài ra, lớp còn có thêm câu lệnh `\addcontentsline{toc}{chapter}{#1}` để đưa “Tài liệu tham khảo” hoặc “Reference” vào mục lục.

Kế đến là gói dùng để định dạng đề mục cho “Danh sách hình ảnh” ([List of Figures, LoF](#)), “Danh sách bảng” ([List of Tables, LoT](#)) và “Mục lục” ([Table of Contents, ToC](#)).

```

155 \usepackage{tocloft} %Configure Table of Content , List of
    Figures , List of Tables

```

Gói `tocloft`, cung cấp công cụ dùng cho định dạng các tiêu đề của **LoF**, **LoT** và **ToC**, người dùng còn có thể định nghĩa danh sách mới với gói này [20]. Lưu ý, đây là gói không có sẵn trong TeX Live cơ bản. Công dụng của gói này thoát nhìn có vẻ sẽ bị xung đột với gói `titlesec`, nhưng do tiêu đề của ba đối tượng trên không chịu ảnh hưởng của gói đó (do chúng vốn là các môi trường riêng biệt), nên việc sử dụng `tocloft` vẫn là cần thiết.

```

157 \renewcommand{\cfttoctitlefont}{\hfill\LARGE\bfseries}
158 \renewcommand{\cftaftertoctitle}{\hfill}
159 \renewcommand{\cftloftitlefont}{\hfill\LARGE\bfseries}
160 \renewcommand{\cftafterloftitle}{\hfill}
161 \renewcommand{\cftlottitlefont}{\hfill\LARGE\bfseries}
162 \renewcommand{\cftafterlottitle}{\hfill}

```

Đoạn câu lệnh trên định nghĩa lại hai macro tiêu biểu của ba danh sách **LoF**, **LoT**, **ToC** đó là `\cftXtitlefont` và `\cftafterXtitle` (trong đó X là `toc`, `lof` hoặc `lot`), với `\cftXtitlefont` lưu trữ các câu lệnh định dạng như `font`, kiểu chữ cho tiêu đề và `\cftafterXtitle` lưu trữ câu lệnh mà ta muốn thực hiện ngay sau khi LaTeX đặt tiêu đề. Để thay đổi giá trị hai macro đó ta sử dụng `\renewcommand` của LaTeX.

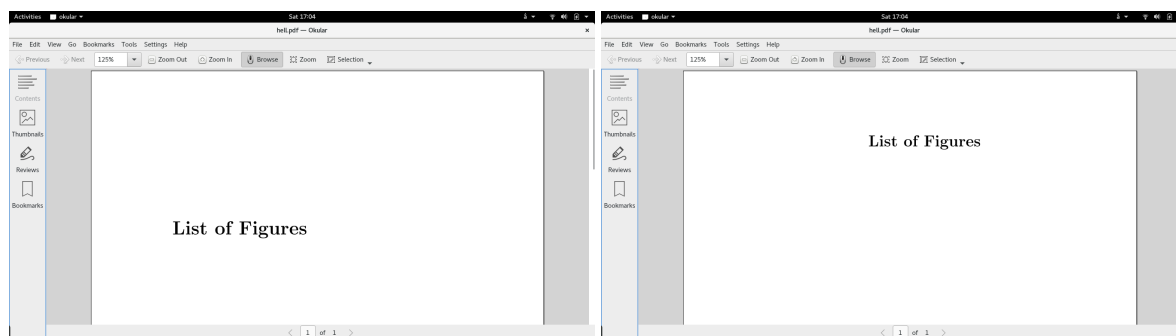
Ở trường hợp này, tôi định nghĩa lại macro `\cftXtitlefont` để lưu trữ câu lệnh `\hfill` dùng để dồn đối tượng sau câu lệnh này về bên phải, kèm theo đó là `\LARGE` để thu nhỏ tiêu đề so với `\Huge` mặc định và `\bfseries` để tô đậm, tiếp theo là macro `\cftafterXtitle` lưu câu lệnh `\hfill` để kết hợp với `\hfill` của `\cftXtitlefont` nhằm đưa tiêu đề ra giữa. Tiếp theo là các macro về khoảng cách:

```

164 \setlength{\cftbeforeloftitleskip}{-15pt}
165 \setlength{\cftbeforelottitleskip}{-15pt}
166 \setlength{\cftbeforetoctitleskip}{-15pt}

```

Các macro `\cftbeforeXtitleskip` lưu trữ giá trị khoảng cách giữa lề đầu với tiêu đề của danh sách, ta thay đổi giá trị của chúng bằng câu lệnh `\setlength`. Trong trường hợp này, giá trị là số âm với lí do tương tự như với câu lệnh `\titlespacing*`. Kết hợp các câu lệnh trên, kết quả cho ra khi sử dụng câu lệnh `\listoffigures` (là câu lệnh có chức năng tự tổng hợp và lập **LoF** của LaTeX, ta cũng có `\listoftables` và `\tableofcontents` cho **LoT** và **ToC**) được trình bày ở hình 3.4.

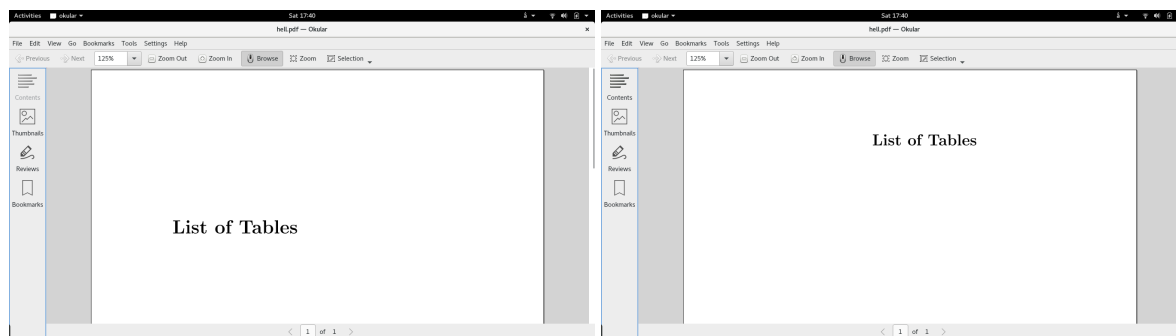


(a) Sử dụng lớp book

(b) Sử dụng lớp vlththesis

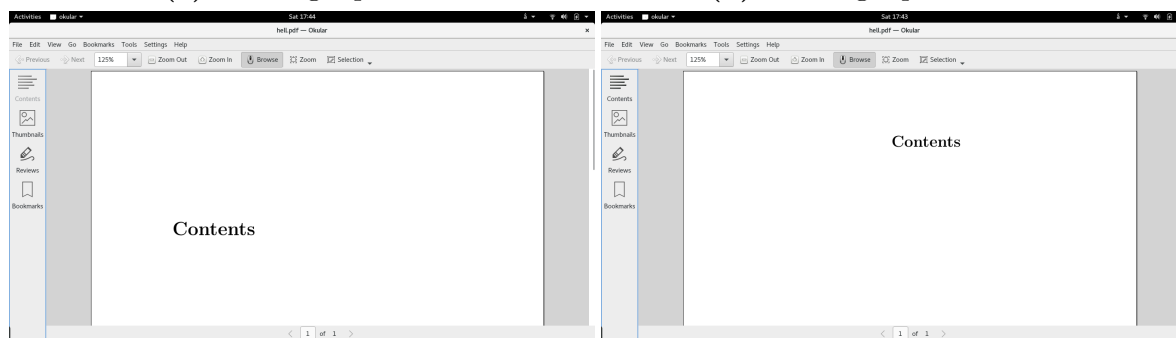
Hình 3.4: So sánh tiêu đề danh sách hình vẽ của hai lớp văn bản

Điều tương tự cũng xảy ra khi sử dụng `\listoftables` và `\tableofcontents`:



(a) Sử dụng lớp book

(b) Sử dụng lớp vlththesis



(c) Sử dụng lớp book

(d) Sử dụng lớp vlththesis

Hình 3.5: So sánh tiêu đề danh sách bảng và mục lục của hai lớp văn bản

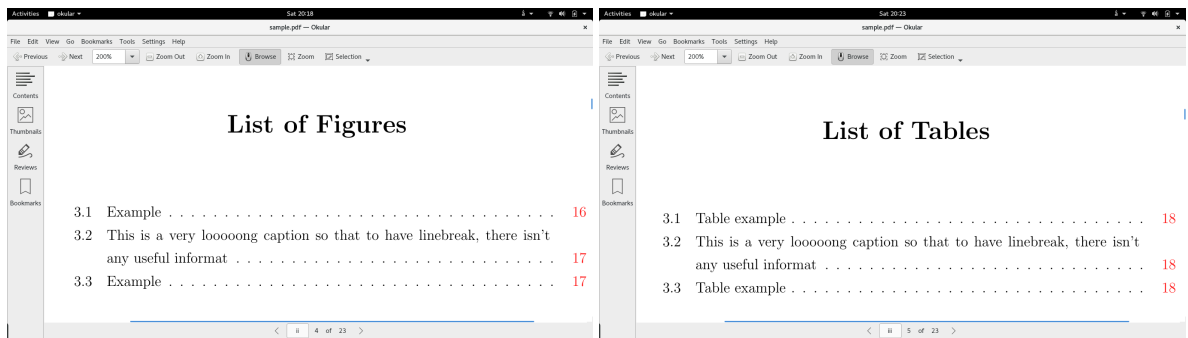
Tiếp theo là định dạng dành cho thành phần trong danh sách hình và bảng.

```

168 \renewcommand\cftfigpresnum{\bfseries\figurename~}
169 \newlength\mylength
170 \settowidth\mylength{\cftfigpresnum}
171 \addtolength\cftfignumwidth{\mylength}
172
173 \renewcommand\cfttabpresnum{\bfseries\tablename~}
174 \settowidth\mylength{\cfttabpresnum}
175 \addtolength\cfttabnumwidth{\mylength}

```

Trước hết ta xét cách trình bày mặc định các thành phần trong danh sách hình vẽ và bảng nếu không có các câu lệnh trên ở hình 3.6:

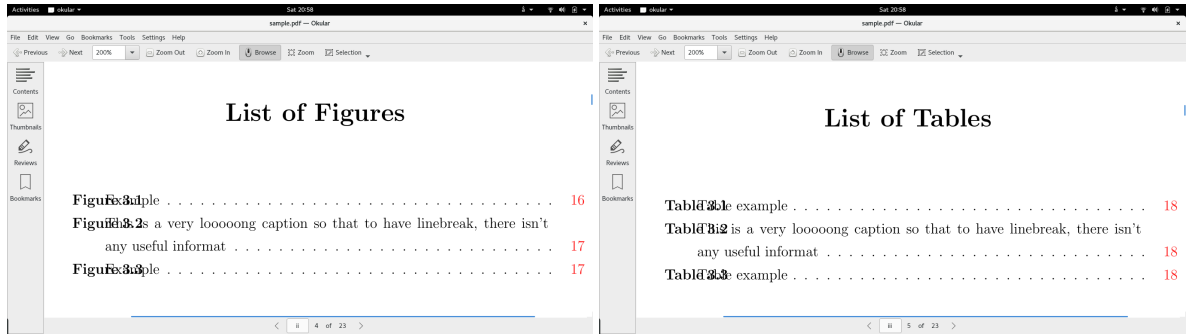


(a) Danh sách hình vẽ

(b) Danh sách bảng

Hình 3.6: Định dạng mặc định của LaTeX dành cho thành phần danh sách

Như ta có thể thấy, các hình chỉ có nhãn là số hiệu, để thêm chữ “Hình”, “Bảng” (hoặc “Figure”, “Table”,... tùy theo ngôn ngữ khai báo trong `babel`), các giá trị đó, nếu có, sẽ được lưu trong macro `\cftZpresnum` của `tocloft` (Z là `fig` hoặc `tab` tương ứng với hình và bảng). Bên cạnh đó, `\figurename`, `\tablename` là hai macro hệ thống LaTeX dùng để lưu giữ nhãn của hình và bảng bằng nhiều ngôn ngữ khác nhau (được sử dụng làm nhãn cho caption). Do đó, bằng cách sử dụng câu lệnh `\renewcommand`, ta có thể dùng macro của LaTeX để định nghĩa `\cftZpresnum` vốn đang bị trống theo mặc định. Sau khi bổ sung nhãn đó ta lại xuất hiện hiện tượng nhãn và caption bị chồng lên nhau như ở hình 3.7.

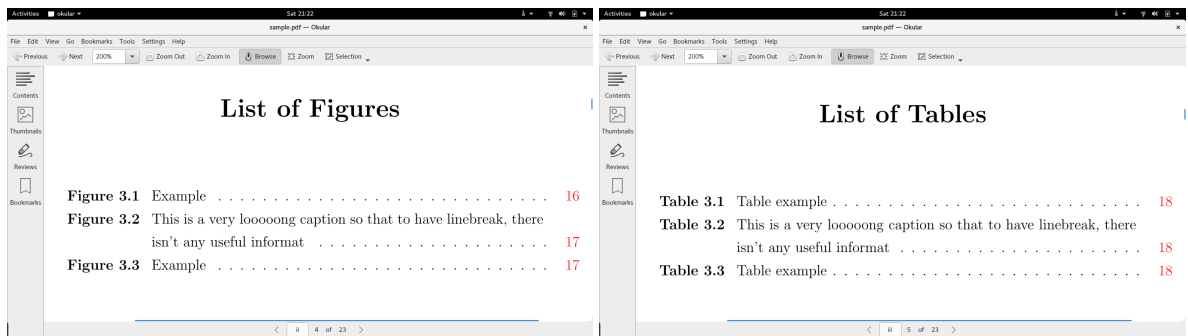


(a) Danh sách hình vẽ

(b) Danh sách bảng

Hình 3.7: Nội dung và nhãn bị chồng nhau sau câu lệnh bổ sung

Để khắc phục điều này, ta cần thay đổi khoảng cách giữa nhãn và tiêu đề, được lưu trong macro `\cftZnumwidth`, bằng cách tăng khoảng cách cũ bằng đúng với chiều dài của nhãn. Điều đó được thực hiện bằng các câu lệnh ở dòng 169-171 và 174-175. Sau khi triển khai các câu lệnh đó ta được kết quả như hình 3.8.



(a) Danh sách hình vẽ

(b) Danh sách bảng

Hình 3.8: Kết quả mong muốn của lớp *vlththesis.cls*

Khai báo tiếp theo là cho gói tạo danh sách kí hiệu viết tắt.

```
177 \usepackage[acronym]{glossaries} %Construct List of Acronyms
```

Gói này cung cấp các câu lệnh giúp ta định nghĩa chữ viết tắt, các thuật ngữ, sử dụng tùy chọn **acronym** cho phép ta tiếp cận các câu lệnh dành cho việc định nghĩa, tạo liên kết, sắp xếp, xây dựng và in danh sách các kí hiệu viết tắt. Để tạo danh sách này, ở phần tiền tố, người dùng sử dụng câu lệnh `\makenoidxglossaries` (không phải là `\makeglossaries`, tham khảo tài liệu [19] để biết thêm chi tiết), sau đó dùng câu lệnh `\newacronym{nhãn}{chữ viết tắt}{nghĩa đầy đủ}` để định nghĩa chữ viết tắt như ví dụ sau:

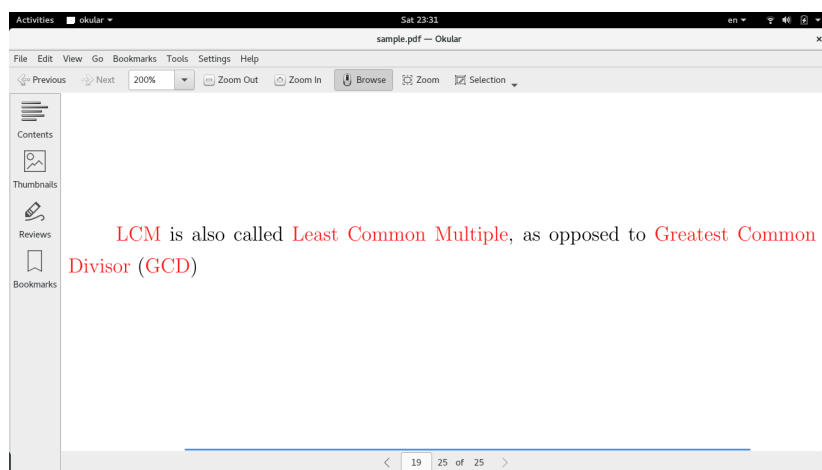
```
\makenoidxglossaries
```

```
\newacronym{gcd}{GCD}{Greatest Common Divisor}
```

```
\newacronym{lcm}{LCM}{Least Common Multiple}
```

Sau khi đã định nghĩa, người dùng được cung cấp ba câu lệnh `\acrshort{nhãn}`, `\acrlong{nhãn}` và `\acrfull{nhãn}` để trình bày chữ viết tắt đã định nghĩa như sau:

```
\acrshort{lcm} is also called \acrlong{lcm}, as opposed to \acrfull{gcd}
```



Hình 3.9: Ví dụ gói *glossaries*

Để in danh sách, ta sử dụng `\printnoidxglossary[type=\acronymtype]` ở vị trí muốn in danh sách các từ viết tắt, người dùng muốn biết thêm các câu lệnh và tùy chọn cho `\printnoidxglossary` có thể tham khảo tài liệu hướng dẫn [19]. Lưu ý rằng để in chữ viết tắt, người dùng phải tham chiếu trong văn bản chữ đó *ít nhất một lần*.

Tiếp theo là ba câu lệnh canh chỉnh cho đoạn văn (paragraph).

```
179 \setlength{\parskip}{0.2cm}
180 \setlength{\parindent}{0.39in}
181 \linespread{1.3}
```

Câu lệnh `\setlength` dùng để định giá trị cho các macro quy định khoảng cách trong đó: `\parskip` là khoảng cách giữa các đoạn văn, `\parindent` là khoảng cách thụt đầu dòng của dòng đầu tiên của các đoạn văn. Câu lệnh `\linespread` dùng để định khoảng cách giữa các dòng trong cùng một đoạn văn, với `\linespread{1.3}`

tương đương 1.5 line trong trình soạn thảo Microsoft Word. Trong ba giá trị trên `\parindent` và `\linespread` dựa trên mẫu báo cáo khoá luận chuẩn của bộ môn.

3.3.3 Các câu lệnh của *vlththesis*

Ngoài tích hợp sẵn các gói và đưa ra các thiết lập mặc định, lớp cũng có các câu lệnh riêng.

- `\supervisorName{text}`

Câu lệnh này cho phép người dùng khai báo tên của cán bộ hướng dẫn (CBHD) ở phần tiền tố, câu lệnh này được định nghĩa như sau:

```
221 \def\thesupervisorName{}
222 \newcommand{\supervisorName}[1]{\def\thesupervisorName{#1}}
```

Câu lệnh này sẽ đẩy tên CBHD vào macro `\thesupervisorName` để sử dụng cho câu lệnh dưới đây.

- `\printcoverpage`

Câu lệnh này được định nghĩa, sử dụng các macro đã được định nghĩa riêng và các câu lệnh trong LaTeX, để thiết kế bố cục cho trang bìa của bài báo cáo theo đúng mẫu chuẩn của khoá luận. Người dùng cần sử dụng các câu lệnh khai báo của LaTeX là `\title{text}` và `\author{text}`, kèm câu lệnh `\supervisorName{text}` của *vlththesis* để cung cấp tiêu đề báo cáo (tên đề tài), tên người thực hiện và CBHD cho câu lệnh sử dụng nhằm xây dựng trang bìa. Trường hợp có nhiều hơn một người thực hiện hay hướng dẫn, người dùng hãy trình bày theo ví dụ sau để đảm bảo lớp trình bày đúng bố cục: `\author{Nguyen Thi A\&Tran Thi B\&Vo Van C}`. Người dùng cũng thực hiện tương tự với câu lệnh `\supervisorName{text}`. Trang bìa của báo cáo này được in ra sử dụng chính câu lệnh trên. Định nghĩa chi tiết của câu lệnh được nêu ở phần phụ lục A.

Ý tưởng câu lệnh này dựa trên câu lệnh tương tự của lớp *gsemthesis* [10].

- `\acknowledgements{text}`

Câu lệnh này dùng để đẩy “Lời cảm ơn” vào macro `\theacknowledgements`, người dùng có thể soạn trực tiếp lời cảm ơn vào giữa *text* của câu lệnh, hoặc soạn riêng một tập tin `.tex` (tập tin này không cần thiết phải có các câu lệnh LaTeX) cho lời cảm ơn và sử dụng lệnh `\input{file}` để trỏ tập tin đó vào câu lệnh (`\acknowledgements{\input{file}}`).

- `\printfrontmatter`

Đây là câu lệnh dùng để in Lời cảm ơn, Mục lục, Các kí hiệu viết tắt, Danh sách hình vẽ, Danh sách bảng theo đúng thứ tự liệt kê như trên. Đối với Lời cảm ơn, `\printfrontmatter` sử dụng macro `\theacknowledgements`, câu lệnh ẩn số trang (`\pagenumbering{gobble}`), sử dụng `\chapter*` để đặt tiêu đề Lời cảm ơn,... Tiếp theo, câu lệnh tích hợp `\tableofcontents` nhằm in ra Mục lục, để in Danh sách hình vẽ và Danh sách bảng, câu lệnh `\printfrontmatter` sau đó sử dụng `\conditionalLoF` và `\conditionalLoT`, hai biến thể của `\listoffigures` và `\listoftables` được định nghĩa trong lớp này như sau:

```
293 \newcommand\conditionalLoF{\iftotalfigures\listoffigures\fi}
294 \newcommand\conditionalLoT{\iftotaltables\listoftables\fi}
```

Sử dụng hai câu điều kiện `\iftotalfigures...\fi` và `\iftotaltables...\fi` của gói `totalcount`, lớp kiểm tra xem người dùng có chèn hình và bảng vào văn bản (chỉ tính những hình sử dụng môi trường ***figure*** và bảng sử dụng môi trường ***table*** chứa ***tabular***) trước khi quyết định in danh sách.

Câu lệnh sử dụng `\printnoidxglossary[type=\acronymtype,title=\acrtitle,style=listdotted]` để in Các kí hiệu viết tắt, và được kiểm tra bởi câu điều kiện `\iftoggle{noacr}`, lớp cung cấp tùy chọn ***noarc*** để người dùng sử dụng trong trường hợp không muốn in danh sách chữ viết tắt hay hoàn toàn không sử dụng chữ viết tắt nào trong văn bản, khi đó người dùng cần khai báo tùy chọn này để ngăn không cho lớp in danh sách. Từ khoá `title` của câu lệnh `\printnoidxglossary` được truyền vào giá trị là một macro `\acrtitle` với định nghĩa như sau:

```

287 \iftoggle{viet}{%
288     \def\acrttitle{Các kí hiệu viết tắt}%
289 }{%
290     \def\acrttitle{Acronym}%
291 }

```

Câu điều kiện `\iftoggle{viet}` kiểm tra tùy chọn **vietnamese** của lớp nhằm định nghĩa giá trị thích hợp cho macro. Thêm vào đó, câu lệnh tích hợp câu `\frontmatter` và `\mainmatter` của book để đánh số trang la mã cho các đối tượng trên và trả về số thường cho các chương chính. Định nghĩa đầy đủ của câu lệnh này được liệt kê trong phụ lục A. Lời cảm ơn, Mục lục, Các kí hiệu viết tắt, Danh sách hình, bảng của báo cáo này được tổng hợp và in ra tự động sử dụng duy nhất một câu lệnh này.

• `\startintroduction`

Đây là câu lệnh dùng để thay thế cho `\chapter*{Lời giới thiệu}` hay `\chapter*{Introduction}`, định nghĩa của câu lệnh này như sau:

```

338 \newcommand{\startintroduction}{%
339     \iftoggle{viet}{
340         \chapter*{Lời giới thiệu}
341         \markboth{}{Lời giới thiệu}
342         \addcontentsline{toc}{chapter}{Lời giới thiệu}%
343     }{%
344         \chapter*{Introduction}
345         \markboth{}{Introduction}
346         \addcontentsline{toc}{chapter}{Introduction}%
347     }\label{ch:intro}
348 }
349 }

```

Ta có thể thấy, câu lệnh này kèm theo `\iftoggle{viet}` để kiểm tra tùy chọn của lớp nhằm đưa giá trị thích hợp làm tiêu đề. Câu `\addcontentsline` để đưa Lời giới thiệu vào mục lục (mặc định LaTeX không đưa các chương không đánh số vào mục lục) và `\markboth` để cập nhật lại header.

Hai câu lệnh `\printfrontmatter` và `\startintroduction` trên cũng lấy ý tưởng

từ hai câu lệnh cùng tên của lớp `gsemthesis` [10].

• `\thebackmatter`

Câu lệnh này dùng để đánh dấu bắt đầu phụ lục, tất cả các `\chapter` sau câu lệnh này đều được hiểu và sẽ được LaTeX dán nhãn phụ lục và đánh số bằng chữ cái, ví dụ “Phụ lục A”, “Phụ lục B”,... Định nghĩa của câu lệnh này như sau:

```

351 \newcommand{\thebackmatter}{%
352     \titleformat{\chapter}[display]
353     {\normalfont\filcenter\Large\bfseries}{\
chaptertitlename\ \thechapter}{20pt}{\LARGE}
354     \titlespacing*{\chapter}{0pt}{-50pt}{40pt}
355     \appendix
356     \pagestyle{supplement}
357 }
```

Hai câu lệnh `\titleformat` và `\titlespacing*` dùng để tinh chỉnh lại đề mục cho phụ lục, loại bỏ các đường kẻ và dời lên đầu trang một khoảng ngắn, phụ lục của báo cáo này là kết quả, đồng thời, câu lệnh định lại style cho các trang sau câu lệnh thành `supplement` (được đề cập ở 3.3.2 dòng lệnh 101-106).

• `\reference`

Câu lệnh này dùng để thay thế cho `\printbibliography` kèm theo các biến đổi sau:

```

359 \newcommand{\reference}{
360     \backmatter
361     \printbibliography
362     \thispagestyle{plain}
363 }
```

Câu `\backmatter` dùng để báo tất cả các câu lệnh chương mục sau câu lệnh này sẽ không được đánh số, đây chỉ là câu lệnh đánh dấu các phần phụ trợ sau phần chính của sách, đồng thời định style `plain` cho Tài liệu tham khảo.

CHƯƠNG 4

Kết luận và hướng phát triển

4.1 Kết luận

Đề tài tập trung tổng hợp các gói, khai báo và xây dựng các câu lệnh để tạo thành một tập tin lớp (class file) hỗ trợ định dạng bố cục cho các khoá luận tốt nghiệp và báo cáo, đề tài cũng đồng thời đưa ra những hướng dẫn cơ bản để người dùng có thể cài đặt và nắm bắt được cách tạo một tập tin đầu vào LaTeX đơn giản, giúp người dùng làm quen với việc chuyển hoá ý tưởng thiết kế thành cấu trúc logic, để tập trung hơn vào nội dung và trình tự sắp xếp các nội dung của văn bản hơn là tốn quá nhiều thời gian cho định dạng bên ngoài, kế thừa đúng với mục đích của những người sáng tạo ra LaTeX.

Tuy chỉ mới thành công trong việc định dạng cơ bản các thành phần của một báo cáo khoá luận, cũng như chưa có những câu lệnh chặt chẽ và môi trường mới để giúp người dùng thuận tiện trong việc thiết kế các bố cục khác nhau, do thời gian tìm hiểu có giới hạn, nhưng sản phẩm cũng đủ để trở thành nền tảng cho nhiều hướng phát triển sau này.

Ngoài tập tin lớp, bản thân báo cáo này cũng được xem là một thành phẩm của đề tài, do được tạo ra hoàn toàn nhờ sử dụng lớp `vlththesis` này. Đồng thời, báo cáo cũng đã tổng hợp các tài liệu, nguồn thông tin cần thiết, phân tích đầy đủ cấu trúc của tập tin lớp, nhằm giúp những ai quan tâm có thể dễ dàng nắm bắt, chỉnh sửa, bổ sung, phát triển thêm cho lớp và sử dụng được thêm đầy đủ tính năng của các g tích hợp.

4.2 Hướng phát triển của đề tài

Do tính chất mã nguồn mở của LaTeX, số lượng gói và lớp hỗ trợ LaTeX ngày càng được phát triển thêm nhiều tính năng, bố cục, môi trường mới tối ưu hơn những gói cũ, việc tìm hiểu, cập nhật các gói mới và tối ưu hơn để tích hợp vào tập tin lớp

cũng là điều cần thiết. Hơn nữa, do chỉ mới được xây dựng, lớp cần phải trải qua một thời gian dài sử dụng mới có thể biết được những thiếu sót cần bổ sung, vì vậy việc tham khảo thêm các lớp mới, tìm kiếm gói phù hợp bù đắp thiếu sót cũng là một hướng phát triển¹.

Như đã nói ở trên, lớp vẫn còn thiếu những câu lệnh giúp người dùng linh hoạt hơn trong việc thiết kế bố cục, hiện nay, lớp chỉ dừng ở mức cho ra văn bản theo đúng khuôn khổ định trước mà chưa thực sự cho phép người soạn thảo tham gia vào quá trình thiết kế, tuy có thể thay đổi những thiết lập gói mặc định của lớp, nhưng nếu muốn thay đổi thứ tự các thành phần, người dùng cần phải thay đổi trong tập tin lớp, hoặc định nghĩa lại hoàn toàn câu lệnh, do đó cần phải tìm hiểu thêm nhiều thủ thuật lập trình LaTeX, để phát triển thêm các câu lệnh linh hoạt hơn và các môi trường trình bày thông tin mới. Trước mắt có thể phát triển cho lớp hỗ trợ tốt hơn cho việc tạo các văn bản in hai mặt giấy.

Đề tài còn có thể phát triển thêm để tìm hiểu sâu hơn về TeX, các macro và câu lệnh của nó. Thao tác với các câu lệnh và macro ở mức TeX, sẽ giúp cho câu lệnh chặt chẽ và hạn chế lỗi nhiều hơn, tuy nhiên, do đây là nền tảng cấp thấp hơn LaTeX, nên các cú pháp của câu lệnh khá khó đọc và chuyên sâu, đòi hỏi nhiều thời gian tìm hiểu, nghiên cứu.

Trong số các gói cần tìm hiểu, đáng chú ý nhất là **Tikz**, cho phép người dùng tạo (vẽ) các đối tượng hình thể trong LaTeX (graphic element) như: đường thẳng, hình tròn, hình chữ nhật, đường cong,... Đây là một gói mạnh mẽ và phức tạp hỗ trợ đắc lực cho các công việc thiết kế, lập đồ thị và nhiều ứng dụng khác, việc tìm hiểu câu lệnh của gói này và tích hợp vào **vlththesis** sẽ mở ra các khả năng cho phép người dùng có thêm nhiều lựa chọn trang trí, thiết kế khung viền, hay lập đồ thị, sơ đồ khối,...

Vấn đề font chữ vẫn chưa được đề cập trong đề tài này, hướng phát triển tiếp theo có thể tập trung vào tìm hiểu thêm về các khái niệm bộ mã kí tự và các gói về font chữ để hạn chế những cảnh báo giải mã tiếng Việt còn tồn đọng trong lớp. Các khái niệm trong LaTeX, như hộp (box) và khoảng cách (length) cũng nên được tìm hiểu nhằm phát triển nhiều hiệu ứng cho các kí tự, hiện nay có nền tảng biến thể XeLaTeX và XeTeX tập trung về vấn đề font chữ này.

Các gói và lớp hiện nay được phát triển sử dụng ba công cụ được tích hợp trong các gói phân phối đó là lớp **ltxdoc**, gói **doc** và công cụ **docstrip**, các công cụ này giúp

¹Tuy nhiên vẫn cần phải chú ý tới độ tương thích của chúng để tránh xung đột khi tích hợp quá nhiều

người viết có thể xây dựng lớp, gói và tạo văn bản hướng dẫn, thông tin trong cùng một tập tin duy nhất, phần văn bản vừa đóng vai trò chú thích cho câu lệnh, vừa trở thành câu chữ trong văn bản hướng dẫn khi được xử lý trực tiếp bằng trình soạn thảo hỗ trợ LaTeX, khi được xử lý qua công cụ `docstrip`, tập tin tích hợp đó sẽ trở thành một tập tin `.sty` hay `.cls` thông thường [4]. Các công cụ trên cung cấp người viết phương tiện để xây dựng các gói lớn, giúp cho họ dễ bảo trì, phát triển, soạn thảo văn bản hướng dẫn trong cùng một loại tập tin mà không cần tạo nhiều tập tin riêng với các tên mở rộng khác nhau cho các mục đích trên. Do quy mô của lớp `vlththesis` và thời gian cho phép có hạn, nên đề tài chưa có cơ hội tìm hiểu thêm và ứng dụng các công cụ đó để xây dựng tập tin lớp này, việc tìm hiểu về `doc` và `docstrip` là hướng phát triển cần thiết cho những ai muốn xây dựng các gói và lớp quy mô, có thể phân phối được trên CTAN.

Và cuối cùng là tìm hiểu thêm về cách kết nối với cộng đồng người dùng LaTeX, ứng dụng GitHub và CTAN để phân phối sản phẩm, nhận đánh giá và đóng góp từ người dùng, hướng đến việc xây dựng, bảo trì lớp và gói sản phẩm theo đúng tinh thần của cộng đồng mã nguồn mở, tìm hiểu kĩ càng về giấy phép LPPL để có hướng phát triển đúng đắn cho thành phẩm.

Phụ lục A

Định nghĩa đầy đủ của câu lệnh

Định nghĩa đầy đủ của hai câu lệnh `\printcoverpage` và `\printfrontmatter` ở phần 3.3.3 sẽ được nêu chi tiết ở phần này. Như đã nói ở mục đó, `\printcoverpage` sử dụng các macro được định nghĩa riêng trong lớp và các câu lệnh để xây dựng trang bìa. Dưới đây là phần định nghĩa của các macro nói trên.

```
183 \iftoggle{viet}{%
184     \def\theGroup{ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH}%
185 }{%
186     \def\theGroup{Vietnam National University - Ho Chi Minh
187     City}%
188 }
189 \iftoggle{viet}{%
190     \def\theUniversity{TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN}%
191 }{%
192     \def\theUniversity{University of Science}%
193 }
194
195 \iftoggle{viet}{%
196     \def\theFaculty{KHOA VẬT LÝ - VẬT LÝ KỸ THUẬT}%
197 }{%
198     \def\theFaculty{Faculty of Physics and Engineering
199     Physics}%
200 }
201 \iftoggle{viet}{%
202     \def\theDepartment{CHUYÊN NGÀNH VẬT LÝ TIN HỌC}%
203 }{%
204     \def\theDepartment{Department of Physics and Computer
205     Science}%
206 }
```

```

206
207 \iftoggle{viet}{%
208     \def\theReport{KHOÁ LUẬN TỐT NGHIỆP}%
209 }{%
210     \def\theReport{BACHELOR THESIS}%
211 }
212
213 \iftoggle{viet}{%
214     \def\thePlace{TP. HỒ CHÍ MINH}%
215 }{%
216     \def\thePlace{HO CHI MINH CITY}%
217 }
218
219 \def\theDate{\the\year}

```

Các câu lệnh điều kiện `\iftoggle{viet}` dùng để kiểm tra tùy chọn **vietnamese**. Tiếp theo, là định nghĩa đầy đủ của `\printcoverpage`:

```

223 \newcommand{\printcoverpage}{%
224     \thispagestyle{empty}
225     \begin{center}
226     {\bfseries\parskip=0pt
227
228     \theGroup
229     \vspace*{0.1cm}
230
231     \theUniversity
232     \vspace*{0.1cm}
233
234     \theFaculty
235     \vspace*{0.1cm}
236
237     \theDepartment\\
238     \vspace*{0.1cm}
239     -----oOo-----
240     }
241     \vspace*{1cm}
242

```

```

243      {\bfseries
244      \large
245      \theReport}
246
247      \vspace*{2cm plus 1cm minus 0.5cm}
248
249      \iftoggle{viet}
250      {\begin{flushleft}
251          \textsl{\Large\underline{Đề tài:}}
252      \end{flushleft}}
253      {\sim}
254      {\huge\bfseries
255          \@title\par
256      }
257      \end{center}
258
259      \vspace*{2cm plus 1 cm minus 0.5cm}
260
261      \hfill
262      {\bfseries\large
263      \iftoggle{viet}{%
264      \begin{tabular}{r l}
265      \underline{SVTH}: & \@author\\
266      \underline{CBHD}: & \thesupervisorName\\
267      \end{tabular}%
268      }{%
269      \begin{tabular}{r l}
270      \underline{Student}: & \@author\\
271      \underline{Supervisor}: & \thesupervisorName\\
272      \end{tabular}%
273      }}
274
275      \vfill
276      \begin{center}
277      -----\\
278      \bfseries
279      \thePlace\ - \theDate
280      \end{center}

```

```

281 \clearpage
282 }

```

Kế đến là định nghĩa đầy đủ của câu lệnh `\printfrontmatter`:

```

296 \newcommand{\printfrontmatter}{%
297     \frontmatter
298     \fancyhead[L]{}
299     \ifdefempty{\theacknowledgements}{~}{
300     \pagenumbering{gobble}
301     \iftoggle{viet}{%
302     \chapter*{Lời cảm ơn}\markboth{}{Lời cảm ơn}%
303     }{%
304     \chapter*{Acknowledgements}\markboth{}{
Acknowledgements}%
305     }
306     \label{ch:ack}
307     \theacknowledgements
308     \clearpage
309     \pagenumbering{roman}
310     }
311
312     {
313     \tableofcontents
314     }
315
316     \iftoggle{noacr}{~}
317     {%
318     \clearpage
319
320     \printnoidxglossary[type=\acronymtype,title=\
acrtitle, style=listdotted]
321     \addcontentsline{toc}{chapter}{\acrtitle}
322     }
323
324     {%
325     \clearpage
326     \conditionalLoF

```

```
327         }
328
329         {%
330         \clearpage
331         \conditionalLoT
332         }
333
334         \mainmatter
335         \fancyhead[L]{\slshape\nouppercase{\
chaptertitlename\ \thechapter}}
336     }
```


Phụ lục B

Trích đoạn tập tin đầu vào L^AT_EX của đề tài

Mục này bao gồm các trích đoạn của tập tin đầu vào LaTeX của báo cáo này, đóng vai trò như ví dụ sử dụng cho file class sản phẩm. Dưới đây là tập tin `main.tex`.

```
1 \documentclass[vietnamese]{vlththesis}
2
3 \graphicspath{ {image/} }
4 \addbibresource{bibbi.bib}
5 \makenoidxglossaries
6
7 \newacronym{lppl}{LPPL}{LaTeX Project Public License}
8 \newacronym{tex}{\TeX}{Tau Epsilon Chi}
9 \newacronym{wysiwyg}{WYSIWYG}{What you see is what you get}
10 \newacronym{ctan}{CTAN}{Comprehensive \TeX\ Archive Network}
11 \newacronym{ams}{AMS}{American Mathematical Society}
12 \newacronym{isbn}{ISBN}{International Standard Book Number}
13 \newacronym{lof}{LoF}{List of Figures}
14 \newacronym{lot}{LoT}{List of Tables}
15 \newacronym{toc}{ToC}{Table of Contents}
16
17 \title{THIẾT KẾ LUẬN VĂN, BÀI BÁO CÁO SỬ DỤNG HỆ THỐNG LATEX}
18 \author{Trịnh Tích Thiện}
19 \supervisorName{TS. Nguyễn Chí Linh}
20
21 \begin{document}
22 \printcoverpage
23 \acknowledgements{\input{chapter/acknowledgements}}
24 \printfrontmatter
25
26 \include{chapter/introduction}
27 \fancyhead[L]{\slshape\nouppercase{\chaptertitlename\
  thechapter}}
```

```

28 \include{chapter/chapter1}
29 \include{chapter/chapter2}
30 \include{chapter/chapter3}
31 \include{chapter/chapter4}
32 \thebackmatter
33 \include{chapter/appendixA}
34 \include{chapter/appendixB}
35
36 \reference
37 \end{document}

```

Đối với các văn bản phức tạp, ta nên phân các chương hồi ra nhiều tập tin `.tex` khác nhau, sau đó sử dụng câu lệnh `\include` hoặc `\input` (lưu ý khác với `\input`, `\include` sẽ sử dụng `\clearpage` trước khi tích hợp tập tin). Việc này giúp ta dễ dàng kiểm soát nội dung của từng chương và cô lập lỗi phát sinh.

Dưới đây là trích đoạn tập tin `acknowledgements.tex`.

- 1 Đầu tiên, con xin gửi lời biết ơn đến mẹ, người đã thay thế vai trò người cha đã mất, cang đáng cả gia đình và nuôi dưỡng con nên người, con cũng xin cảm ơn dì Chính, người mà con vẫn luôn coi như người mẹ thứ hai, chăm sóc con từng miếng ăn, giấc ngủ và luôn coi con như con ruột của mình, công ơn của hai mẹ dành cho con không từ ngữ nào mà diễn tả được. \par
- 2 Em xin cảm ơn các thầy cô khoa Vật Lý - Vật Lý Kỹ Thuật, đã tận tâm truyền đạt kiến thức cho em trong những năm đầu đại học. Em xin chân thành cảm ơn thầy cô của Bộ môn Vật Lý Tin Học, đã xây dựng bộ môn với các trang thiết bị hiện đại và sự nhiệt tình, thân thiện của các thầy cô, giúp em có thể thoải mái học tập, nghiên cứu mà không cảm thấy căng thẳng, áp lực. Những lời chỉ bảo của thầy cô đã cho em những kiến thức cần thiết và quý báu cho định hướng của mình. \par

Tiếp theo là trích đoạn tập tin `introduction.tex`.

- 1 \startintroduction
- 2 \fancyhead[L]{}
- 3 Ngày nay, ngoài các trình soạn thảo văn bản phổ biến, \LaTeX cũng là một sự lựa chọn dành cho người soạn thảo được tạo ra với triết lý hoàn toàn khác biệt so với các trình hiện hành. Nhận thấy hạn chế của chất lượng in ấn

lúc bấy giờ, và việc người dùng tốn quá nhiều thời gian để định dạng thay vì tập trung soạn thảo, Donald E. Knuth đã phát triển hệ thống TeX, và từ đó, Leslie Lamport xây dựng thành LaTeX, với mục đích giúp người dùng sử dụng câu lệnh để việc thiết kế văn bản được thực hiện một cách tự động bởi hệ thống. \par

Và cuối cùng là trích đoạn của tập tin `chapter1.tex`, các chapter và phụ lục (appendix) cũng có kiểu trình bày ban đầu tương tự.

- 1 \chapter{Tổng quan về \LaTeX}\label{ch:1}
- 2 Khác với quan niệm của nhiều người, LaTeX \emph{không phải} là chương trình hay hệ thống xử lý văn bản (document processor). Trong chương này, ta sẽ nói về định nghĩa và lịch sử của LaTeX và cả TeX, nền tảng mà nó dựa trên.\par
- 3 \section{\LaTeX\ là gì?}\label{sec:1.1}
- 4 LaTeX (phiên âm La-téch hoặc Lây-téch) là hệ thống trình bày văn bản (document preparationsystem)\cite{latex-pro}, được sử dụng phổ biến trong cộng đồng khoa học và học thuật để soạn thảo, trình bày các văn bản khoa học, kĩ thuật. Hiện nay, các nhà khoa học trao đổi tài liệu của họ dưới dạng mã LaTeX \cite{lamport}. Ngoài lĩnh vực học thuật, LaTeX còn được sử dụng để viết các thư từ doanh nghiệp hoặc cá nhân, tin tức, các bài báo, sách và nhiều lĩnh vực khác.\par

Tài liệu tham khảo

- [1] Leslie Lamport. *LaTeX: A Document Preparation System*. 2nd ed. Addison-Wesley Professional, 1994. 288 pp. ISBN: 0-201-52983-1.
- [2] Donald Arseneau. *The notoccite package*. July 20, 2000. URL: <http://mirror.kku.ac.th/CTAN/macros/latex/contrib/notoccite/notoccite.pdf>.
- [3] Donald Arseneau. *The wrapfig package*. v3.6. Jan. 31, 2003. URL: <http://mirror.kku.ac.th/CTAN/macros/latex/contrib/wrapfig/wrapfig-doc.pdf>.
- [4] Frank Mittelbach and Michel Goossens. *The LaTeX Companion*. 2nd ed. Addison-Wesley Professional, 2004. 1120 pp. ISBN: 0-201-36299-6.
- [5] Axel Sommerfeldt. *Customizing captions of floating environments*. Nov. 2, 2011. URL: <http://mirror.kku.ac.th/CTAN/macros/latex/contrib/caption/caption-eng.pdf>.
- [6] Donald Arseneau. *url.sty version 3.4*. Sept. 16, 2013. URL: <http://mirror.kku.ac.th/CTAN/macros/latex/contrib/url/url.pdf>.
- [7] Johannes L. Braams and Frank Mittelbach. *Standard Document Classes for LaTeX version 2e*. v1.4h. Sept. 29, 2014. URL: <http://mirror.kku.ac.th/CTAN/macros/latex/base/classes.pdf>.
- [8] Jobst Hoffmann. *The Listings Package*. v1.6. June 4, 2015. URL: <http://texdoc.net/texmf-dist/doc/latex/listings/listings.pdf>.
- [9] Alan Jeffrey and Frank Mittelbach. *inputenc.sty*. v1.2c. Mar. 17, 2015. URL: <http://mirror.kku.ac.th/CTAN/macros/latex/base/inputenc.pdf>.
- [10] Emmanuel Rousseaux. *The gsemthesis class*. v0.9.4. Uni Mail, Boulevard du Pont-d'Arve 40, 1205 Genève, Switzerland, Feb. 9, 2015. URL: <http://mirrors.ctan.org/macros/latex/contrib/gsemthesis/gsemthesis.pdf>.
- [11] Javier Bezos. *The titlesec, titleps and titletoc Packages*. v2.10.2. Mar. 21, 2016. URL: <http://mirror.kku.ac.th/CTAN/macros/latex/contrib/titlesec/titlesec.pdf>.
- [12] Dr. Uwe Kern. *Extending LaTeX's color facilities: the xcolor*. v2.12. May 11, 2016. URL: <http://mirror.kku.ac.th/CTAN/macros/latex/contrib/caption/subcaption.pdf>.

- [13] Piet van Oostrum et al. *The multirow, bigstrut, bigdelim packages*. v2.2. Nov. 25, 2016. URL: <http://mirror.kku.ac.th/CTAN/macros/latex/contrib/multirow/multirow.pdf>.
- [14] Philipp Lehman et al. *The biblalex package*. v3.9. Dec. 19, 2017. URL: <http://mirror.kku.ac.th/CTAN/macros/latex/contrib/biblalex/doc/biblalex.pdf>.
- [15] Johannes L. Braams and Javier Bezos. *Babel*. v3.15. Nov. 3, 2017. URL: <http://mirror.kku.ac.th/CTAN/macros/latex/required/babel/base/babel.pdf>.
- [16] Philipp Lehman and Joseph Wright. *The etoolbox package*. v2.5. Nov. 22, 2017. URL: <http://mirror.kku.ac.th/CTAN/macros/latex/contrib/etoolbox/etoolbox.pdf>.
- [17] Frank Mittelbach and David Carlisle. *A new implementation of LaTeX's tabular and array environment*. v2.4d. Apr. 15, 2017. URL: <http://mirror.kku.ac.th/CTAN/macros/latex/required/tools/array.pdf>.
- [18] Piet van Oostrum. *Page layout in LaTeX*. v3.6. June 30, 2017. URL: <http://mirror.kku.ac.th/CTAN/macros/latex/contrib/fancyhdr/fancyhdr.pdf>.
- [19] Nicola L.C. Talbot. *The glossaries package v4.35: a guide for beginners*. Nov. 14, 2017. URL: <http://mirror.kku.ac.th/CTAN/macros/latex/contrib/glossaries/glossariesbegin.pdf>.
- [20] Peter Wilson and Herries Press. *The tocloft package*. v2.3i. Aug. 31, 2017. URL: <http://mirror.kku.ac.th/CTAN/macros/latex/contrib/tocloft/tocloft.pdf>.
- [21] *Introduction to LaTeX*. LaTeX3 Project Team. URL: <https://www.latex-project.org/about/>.
- [22] *Learn LaTeX in 30 minutes*. ShareLaTeX team. URL: https://www.sharelatex.com/learn/Learn_LaTeX_in_30_minutes.
- [23] *The Kile Handbook*. Jonathan Pechta et al. URL: <https://docs.kde.org/stable4/en/extragear-office/kile/index.html>.
- [24] *What are TeX and its friends?* CTAN team. URL: <https://www.ctan.org/tex/>.