

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:

THIẾT KẾ LUẬN VĂN, BÀI BÁO  
CÁO SỬ DỤNG HỆ THỐNG  
L<sup>A</sup>T<sub>E</sub>X

SVTH: Trịnh Tích Thiện

CBHD: TS. Nguyễn Chí Linh

—————  
TP. HỒ CHÍ MINH - 2018

# Mục lục

Các kí hiệu viết tắt	ii
Danh sách hình vẽ	iii
Danh sách bảng	iv
<b>1 Tổng quan về L<sup>A</sup>T<sub>E</sub>X</b>	<b>1</b>
1.1 L <sup>A</sup> T <sub>E</sub> X là gì?	1
1.2 Giới thiệu Kile	2
1.3 Lịch sử phát triển của L <sup>A</sup> T <sub>E</sub> X	2
<b>2 Soạn thảo văn bản trong L<sup>A</sup>T<sub>E</sub>X</b>	<b>5</b>
2.1 Cài đặt L <sup>A</sup> T <sub>E</sub> X	5
2.1.1 Cài đặt L <sup>A</sup> T <sub>E</sub> X trên Linux	5
2.1.2 Cài đặt L <sup>A</sup> T <sub>E</sub> X trên Windows	7
2.2 Hướng dẫn sơ lược	14
2.2.1 Đặc điểm của L <sup>A</sup> T <sub>E</sub> X	14
2.2.2 Soạn thảo văn bản	17
<b>3 Thiết kế định dạng văn bản riêng trong L<sup>A</sup>T<sub>E</sub>X</b>	<b>29</b>
3.1 Tổng quan về package và class	29
3.2 Giới thiệu class đề tài	30
3.3 Cấu trúc của class	30
3.3.1 Các khai báo ban đầu	30
3.3.2 Khai báo package và thiết lập	33
<b>Tài liệu tham khảo</b>	<b>36</b>

# Các kí hiệu viết tắt

AMS	.....	American Mathematical Society
CTAN	.....	Comprehensive T <sub>E</sub> X Archive Network
LPPL	.....	LaTeX Project Public License
T <sub>E</sub> X	.....	Tau Epsilon Chi
WYSIWYG	.....	What you see is what you get

# Danh sách hình vẽ

<b>Hình 1.1</b>	Donald E.Knuth, tác giả của TeX . . . . .	<b>2</b>
<b>Hình 1.2</b>	Leslie Lamport, cha đẻ của LaTeX, qua đó giúp phổ biến TeX đến cộng đồng người dùng phổ thông . . . . .	<b>3</b>
<b>Hình 2.1</b>	Kết quả có được sau khi nhập câu lệnh cài Kile . . . . .	<b>5</b>
<b>Hình 2.2</b>	Giao diện trang tải MiKTeX . . . . .	<b>7</b>
<b>Hình 2.3</b>	Chọn tải các package từ mạng . . . . .	<b>8</b>
<b>Hình 2.4</b>	Chọn gói package . . . . .	<b>8</b>
<b>Hình 2.5</b>	Chọn nguồn tải package . . . . .	<b>9</b>
<b>Hình 2.6</b>	Chỉ định đường dẫn tới thư mục chứa package . . . . .	<b>9</b>
<b>Hình 2.7</b>	Quá trình download đang được tiến hành . . . . .	<b>10</b>
<b>Hình 2.8</b>	Chọn tùy chọn cài đặt MiKTeX . . . . .	<b>10</b>
<b>Hình 2.9</b>	Chọn gói MiKTeX cần cài đặt . . . . .	<b>11</b>
<b>Hình 2.10</b>	Chọn quyền truy cập MiKTeX . . . . .	<b>11</b>
<b>Hình 2.11</b>	Trở đường dẫn tới thư mục chứa các package . . . . .	<b>12</b>
<b>Hình 2.12</b>	Chọn thư mục cài đặt . . . . .	<b>12</b>
<b>Hình 2.13</b>	Chọn khổ giấy mặc định và tự động tải . . . . .	<b>13</b>
<b>Hình 2.14</b>	Quá trình cài đặt đang được tiến hành . . . . .	<b>13</b>
<b>Hình 2.15</b>	Kết quả sau khi build file input . . . . .	<b>16</b>
<b>Hình 2.16</b>	Ví dụ một văn bản LaTeX đơn giản . . . . .	<b>18</b>
<b>Hình 2.17</b>	Văn bản đầu ra được định dạng bằng các lệnh cơ bản . . . . .	<b>21</b>
<b>Hình 2.18</b>	Ví dụ cho câu lệnh <code>\emph</code> . . . . .	<b>22</b>
<b>Hình 2.19</b>	Ví dụ xuống dòng . . . . .	<b>23</b>
<b>Hình 2.20</b>	Ví dụ canh vị trí . . . . .	<b>24</b>
<b>Hình 2.21</b>	Ví dụ thêm hình trong văn bản . . . . .	<b>25</b>
<b>Hình 2.22</b>	Ví dụ tạo bảng . . . . .	<b>26</b>
<b>Hình 2.23</b>	Đánh dấu chương và đề mục . . . . .	<b>28</b>

# Danh sách bảng

<b>Bảng 2.1</b>	Các câu lệnh chương mục và thứ bậc . . . . .	28
-----------------	--	----

---

# CHƯƠNG 1

---

## Tổng quan về L<sup>A</sup>T<sub>E</sub>X

Khác với quan niệm của nhiều người, LaTeX *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.

### 1.1 L<sup>A</sup>T<sub>E</sub>X là gì?

LaTeX (phiên âm La-téch hoặc Lây-téch) là hệ thống trình bày văn bản (document preparation system)[13], được sử dụng phổ biến trong cộng đồng khoa học và học thuật để soạn thảo, trình bày các văn bản khoa học, kỹ thuật. Hiện nay, các nhà khoa học trao đổi tài liệu của họ dưới dạng mã LaTeX [1]. 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.

Tính đến nay, LaTeX đã có mặt trên phần lớn các hệ điều hành phổ biến như Windows, Linux, Mac OS,... thông qua các gói phân phối (distribution). Gói phân phối là tập hợp các gói (package) và các chương trình (như trình biên dịch, font chữ, các macro) hỗ trợ người dùng thiết lập môi trường sử dụng LaTeX, mỗi hệ điều hành sẽ có các gói phân phối riêng như TeX Live (Linux), MiKTeX (Windows) hay MacTeX (Mac OS). Bên cạnh hệ điều hành, LaTeX cũng hỗ trợ hoàn toàn hoặc một phần khoảng 200 ngôn ngữ khác nhau, thông qua package đa ngôn ngữ `babel` [10].

LaTeX có nhiều trình soạn thảo (editor) hỗ trợ, từ miễn phí (Kile, TeXmaker, TeXstudio,...), bản quyền (Inlage, Scientific WorkPlace, WinEdt,...) và cả trên nền web ([ShareLaTeX](#), [LaTeX Base](#), [Overleaf](#),...). Ở đề tài này, tôi sử dụng editor Kile trên hệ điều hành Fedora 26, do hệ điều hành và editor này đều là mã nguồn mở.

## 1.2 Giới thiệu Kile

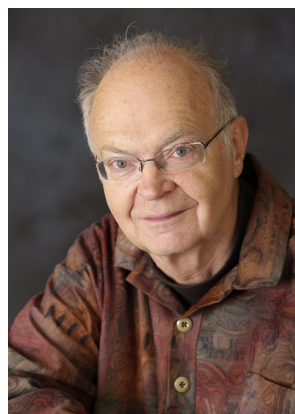
Kile là môi trường LaTeX tích hợp (integrated) dành cho môi trường desktop của KDE, vì vậy nó có thể chạy trên các hệ điều hành UNIX, Linux và cả Windows. Kile cung cấp giao diện đồ hoạ giúp người dùng có thể sử dụng được hết các chức năng của LaTeX một cách trực quan với các công cụ biên dịch, chuyển đổi, debug, xem văn bản sau khi biên dịch, tự động hoàn thành code, thuật sĩ hỗ trợ tạo file LaTeX, tài liệu tham khảo LaTeX kèm theo công cụ quản lý dự án (project management) [15].

Kile được phát triển dựa trên thành phần soạn thảo Kate,... phần lớn các chức năng soạn thảo của Kile là từ Kate. Tuy nhiên, Kile được bổ sung thêm khả năng chỉnh sửa và xử lý văn bản LaTeX.

## 1.3 Lịch sử phát triển của $\text{\LaTeX}$

Dưới đây là lịch sử phát triển của LaTeX và TeX, nền tảng mà LaTeX dựa trên:

- Tháng 5, năm 1977, Donald E.Knuth bắt đầu phát triển một nền tảng xử lý văn bản mới được gọi là “ $\text{\TeX}$  và METAFONT”, mục đích ban đầu của ông khi phát triển TeX là dùng nó để biên soạn sách *The Art of Computer Programming* của mình, sau khi không đồng tình với chất lượng bản in của nhà xuất bản. Dự án **Tau Epsilon Chi** ( $\text{\TeX}$ ) được ông chính thức công bố và khởi động một năm sau đó [4][16].

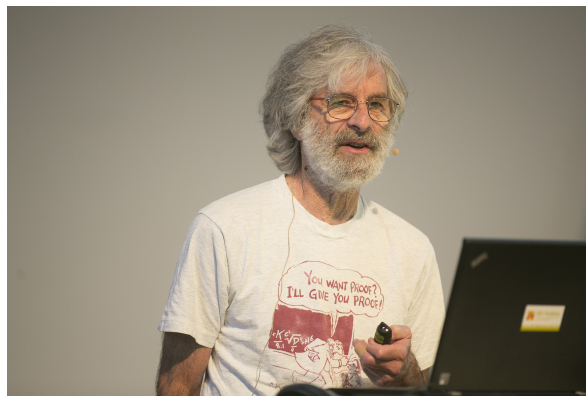


**Hình 1.1:** Donald E.Knuth, tác giả của TeX

- Sau mười năm phát triển, vào thập niên 1990, Donald Knuth chính thức tuyên bố TeX đã hoàn thành phát triển và sẽ không được xây dựng thêm nữa để đảm

bảo tính ổn định. Sau đó, nhiều dự án với mục đích phát triển thêm TeX nở rộ; với một số dự án thành công mở ra thời kì phát triển của sắp xếp chữ tự động (automated typography) [4].

- Từ một dự án phục vụ mục đích cá nhân, TeX còn tạo ảnh hưởng đến Cộng đồng Toán học Mỹ ([American Mathematical Society](#)), được các nhà toán học sử dụng và phát triển thêm, cho ra đời các package ams hỗ trợ nhập liệu công thức, kí hiệu toán học [4].
- Sử dụng engine sắp chữ (typesetting) của TeX và các macro, Leslie Lamport bắt đầu xây dựng một ngôn ngữ mô tả văn bản (document description language) dựa trên hệ thống Scribe của Brian Reid, từ đó cho ra đời LaTeX [4], nền tảng này nhanh chóng chiếm được cảm tình của người dùng do sử dụng các câu lệnh, khai báo cấp cao để "đánh dấu"(markup) văn bản, gần tương tự như các ngôn ngữ markup khác như HTML. LaTeX cho phép người dùng soạn thảo mà không cần tốn quá nhiều thời gian vào việc định dạng (format) và trang trí, sắp xếp văn bản.
- Trong cuốn  *$\text{\LaTeX}$ : A Document Preparation System* có đoạn như sau: “LaTeX là hệ thống sắp xếp văn bản. Phiên bản LaTeX phổ biến đầu tiên, có số hiệu 2.09 một cách bí hiểm<sup>1</sup>, xuất hiện vào 1985” [1]. Bản LaTeX này đã góp phần phổ biến các văn bản TeX ra khỏi cộng đồng toán học Mỹ, nhờ sự ổn định cũng như có đầu tư tốt về tư liệu hướng dẫn.



**Hình 1.2:** Leslie Lamport, cha đẻ của LaTeX, qua đó giúp phổ biến TeX đến cộng đồng người dùng phổ thông

---

<sup>1</sup>Trái với quan niệm “1.0” của nhiều người



- $\text{\LaTeX}$  là ngôn ngữ phổ biến đầu tiên dùng để miêu tả cấu trúc logic của văn bản, giới thiệu triết lý “thiết kế logic” (logical design) đến cộng đồng. Ý nghĩa của “thiết kế logic” là người soạn thảo chỉ nên quan tâm tới thứ tự logic, cấu trúc của văn bản thay vì bận tâm đến hình thức bề ngoài của nó. Tới năm 1994, Lamport đã có thể tự hào viết trong [1] rằng: “ $\text{\LaTeX}$  đã trở nên cực kì phổ biến trong cộng đồng khoa học và học thuật, và được sử dụng rộng khắp trong công nghiệp”, nhưng nếu so với lúc đó, độ phổ biến của  $\text{\LaTeX}$  bây giờ còn lớn hơn rất nhiều đến mức trở nên quen thuộc [4].
- Qua hai hội nghị quan trọng (Exeter UK, 1988 và Karlsruhe Đức, 1989),  $\text{\LaTeX}$  được chấp nhận và sử dụng rộng rãi ở châu Âu, dẫn đến nhu cầu về 1 bản  $\text{\LaTeX}$  quốc tế (International  $\text{\LaTeX}$ ) và các dự án hỗ trợ, chuyển đổi đa ngôn ngữ cho  $\text{\LaTeX}$  của Johannes Braams [4][10].
- Sự thành công của  $\text{\LaTeX}$  kéo theo hàng loạt dự án phát triển cá nhân, làm xuất hiện nhiều biến thể khiến các văn bản  $\text{\TeX}$  gặp vấn đề về tương thích, do được viết trên các biến thể khác nhau. Để giải quyết vấn đề này,  $\text{\LaTeX} 2\epsilon$  ra đời, tích hợp các tính năng cần thiết vào trong lõi (kernel), xây dựng các lớp (class) văn bản tiêu chuẩn, các hỗ trợ thêm về màu, hình ảnh, căn chỉnh vị trí cũng được bổ sung. Bên cạnh đó là cơ chế cho phép người dùng tự xây dựng các tiện ích mở rộng thêm cho  $\text{\LaTeX}$  (thông qua package và class) và đội ngũ hỗ trợ kĩ thuật chuyên nghiệp, liên tục bảo trì kernel  $\text{\LaTeX}$ , các package quan trọng và nhận các báo cáo lỗi từ người dùng.
- Cũng từ đó giấy phép **LaTeX Project Public License (LPPL)** được thành lập để đề ra chuẩn mực và hướng dẫn cho việc phát triển class và package mới, cho phép  $\text{\LaTeX}$  hỗ trợ các mở rộng nhưng vẫn đảm bảo tính ổn định. Hiện nay, các công việc duy trì và phát triển  $\text{\LaTeX}$  được đảm nhiệm bởi  **$\text{\LaTeX}3$  Project Team**. Mục tiêu trước mắt (và lâu dài) của đội ngũ phát triển chính là: tăng cường hỗ trợ đa ngôn ngữ, xây dựng một giao diện thiết kế cho  $\text{\LaTeX}$ , cải thiện quy trình xuất văn bản, cải tiến xử lý định dạng trong đoạn văn, các yêu cầu về font chữ của nhà báo, và tất nhiên, đúng với tên gọi của nhóm, phát triển phiên bản thứ ba cho  $\text{\LaTeX}$ .

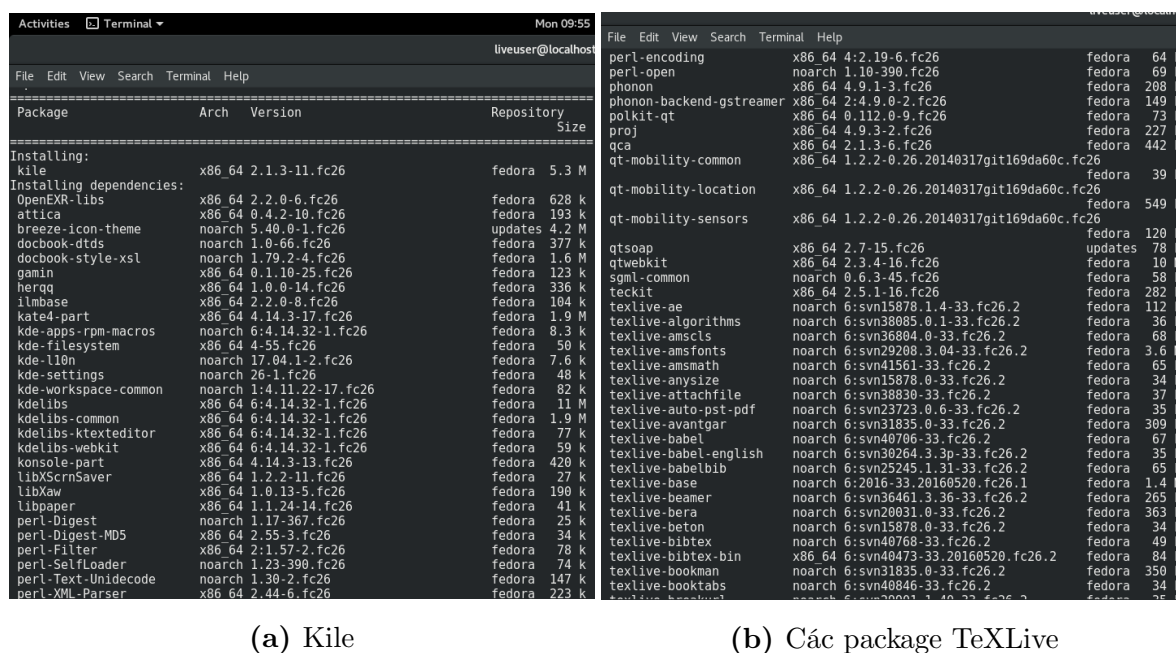
# CHƯƠNG 2

## Soạn thảo văn bản trong $\text{\LaTeX}$

### 2.1 Cài đặt $\text{\LaTeX}$

#### 2.1.1 Cài đặt $\text{\LaTeX}$ 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 package 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 package 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 package 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 package, class tiêu chuẩn. Bộ tài liệu mặc định nằm ở `usr/share/texlive/texmf-dist/doc/latex/base`.

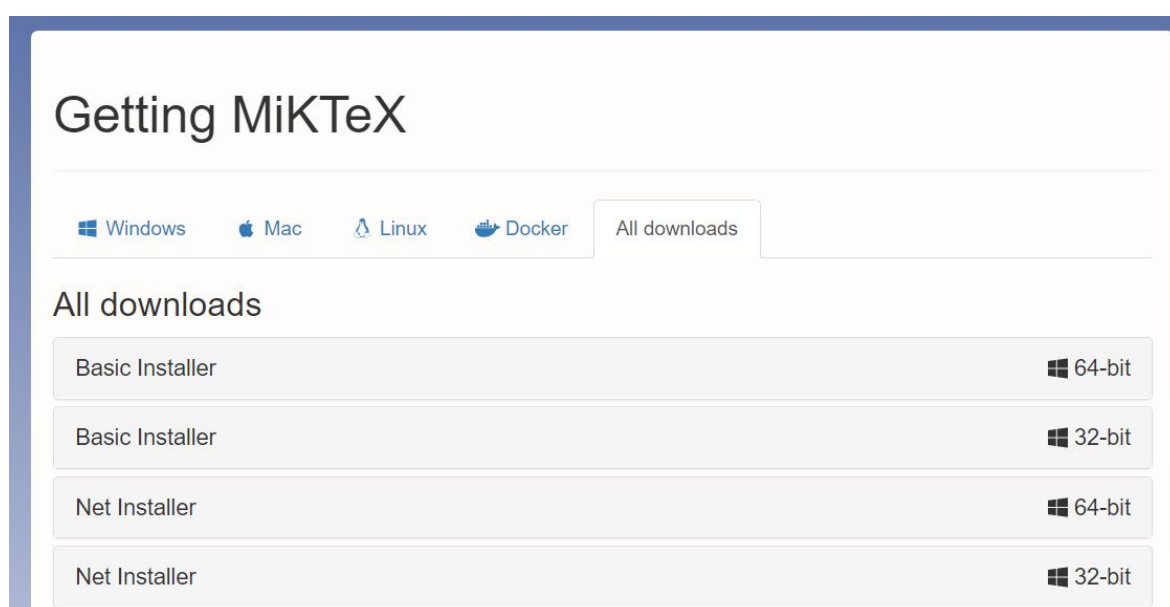
Tương tự, để tải và cài đặt package LaTeX (khi được hệ thống yêu cầu để build file input), người dùng cũng có thể sử dụng `yum` hoặc `apt-get` kèm theo `install texlive-<tên package>`, Linux sẽ tự động tìm kiếm, tải (nếu có) và cài đặt package trên.

### 2.1.2 Cài đặt $\text{\LaTeX}$ trên Windows

Trên nền Windows, ta có thể chọn TeXLive làm distribution 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.

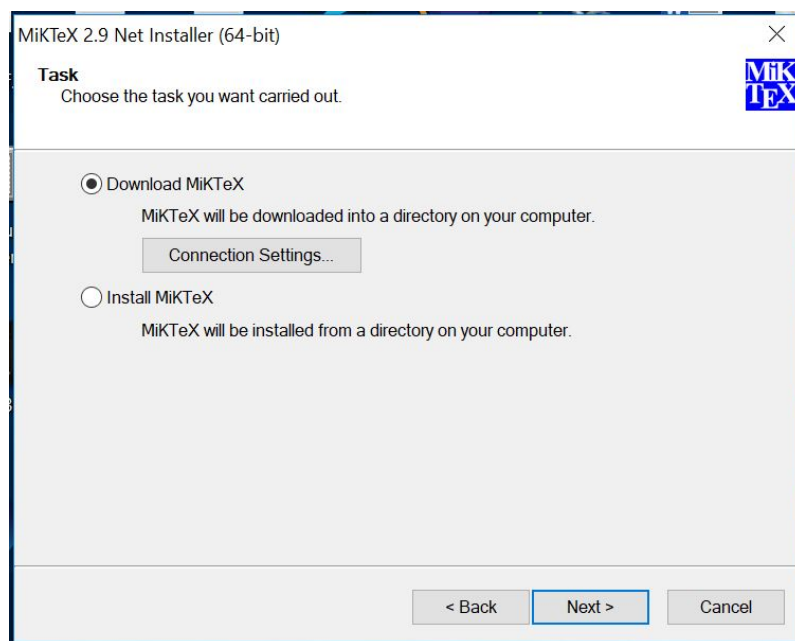
MiKTeX có thể được cài đặt qua các bước sau:

- *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.



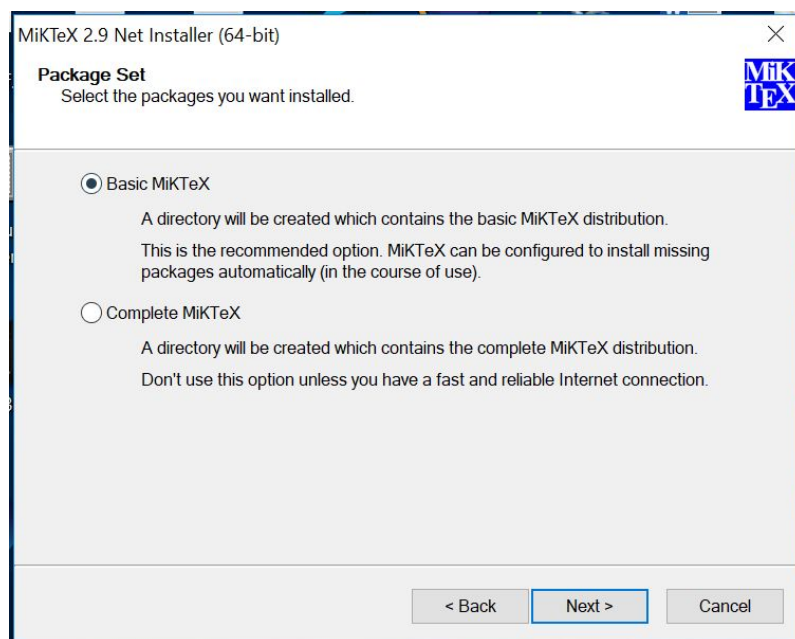
**Hình 2.2:** Giao diện trang tải MiKTeX

- *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 package có sẵn trong máy, hay tải (download) package từ mạng. Do đây là lần cài đặt đầu tiên, ta nên chọn tải package về.



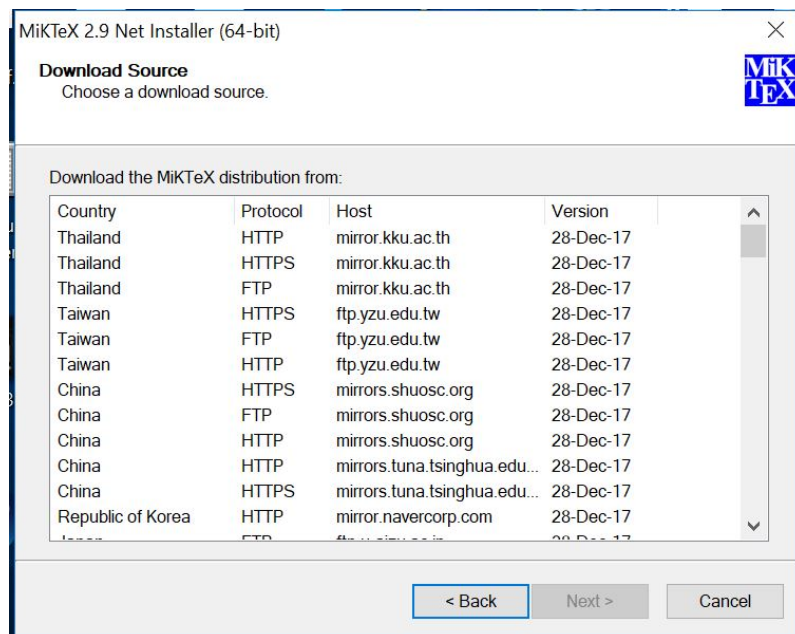
Hình 2.3: Chọn tải các package từ mạng

- *Bước 3:* Ta có hai tùy chọn Basic  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  (các package cơ bản của  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ) hoặc Complete  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  (tất cả các package chính thức của  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ), 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 package còn thiếu, nếu phát hiện trong file văn bản  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  sử dụng package không có sẵn trong máy.



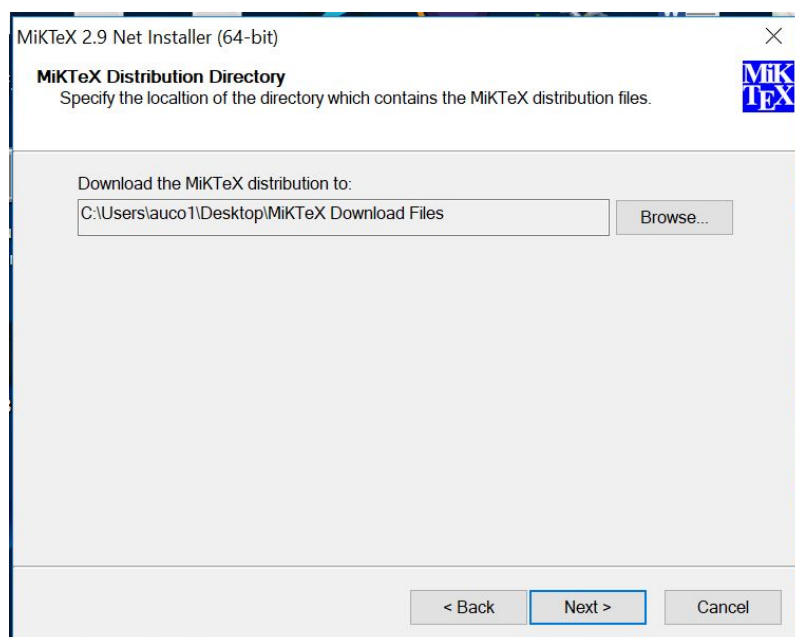
Hình 2.4: Chọn gói package

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



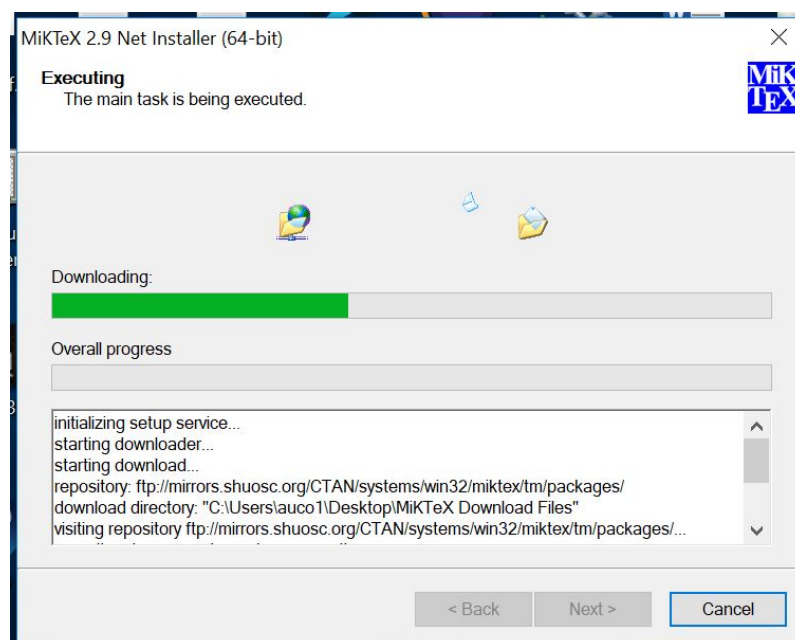
Hình 2.5: Chọn nguồn tải package

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



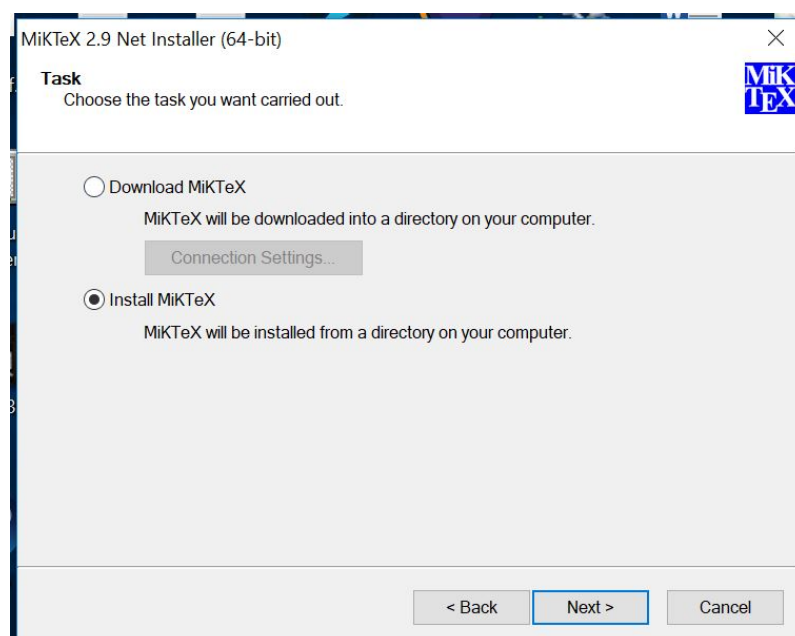
Hình 2.6: Chỉ định đường dẫn tới thư mục chứa package

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 package.



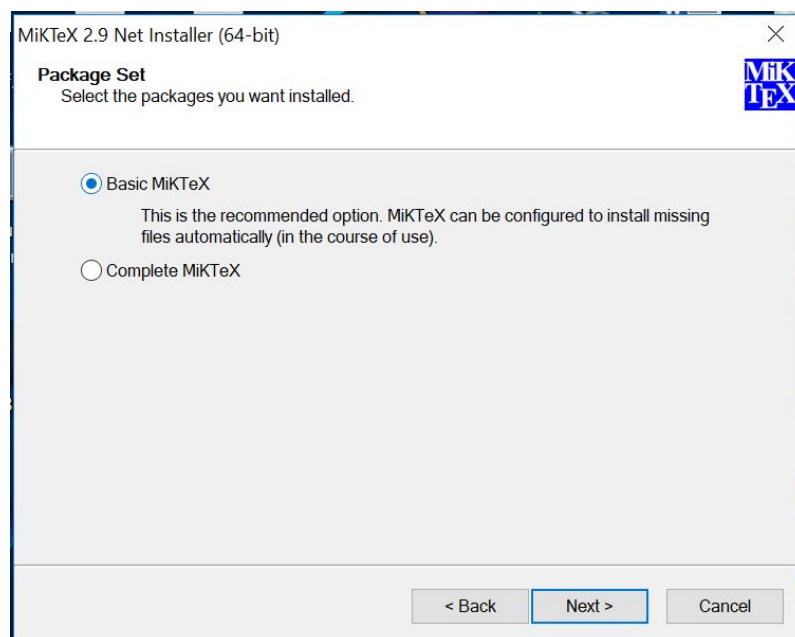
Hình 2.7: Quá trình download đang được tiến hành

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



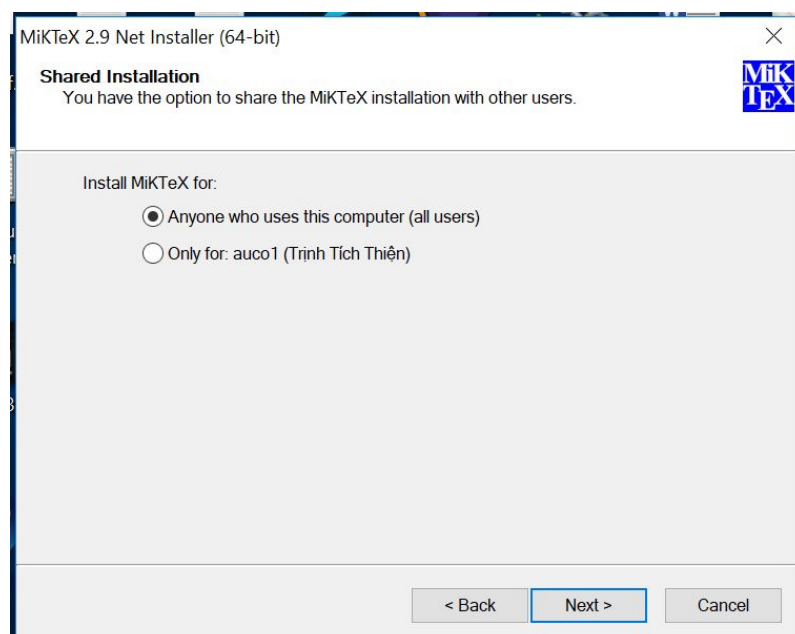
Hình 2.8: Chọn tùy chọn cài đặt MiKTeX

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



Hình 2.9: Chọn gói MiKTeX cần cài đặt

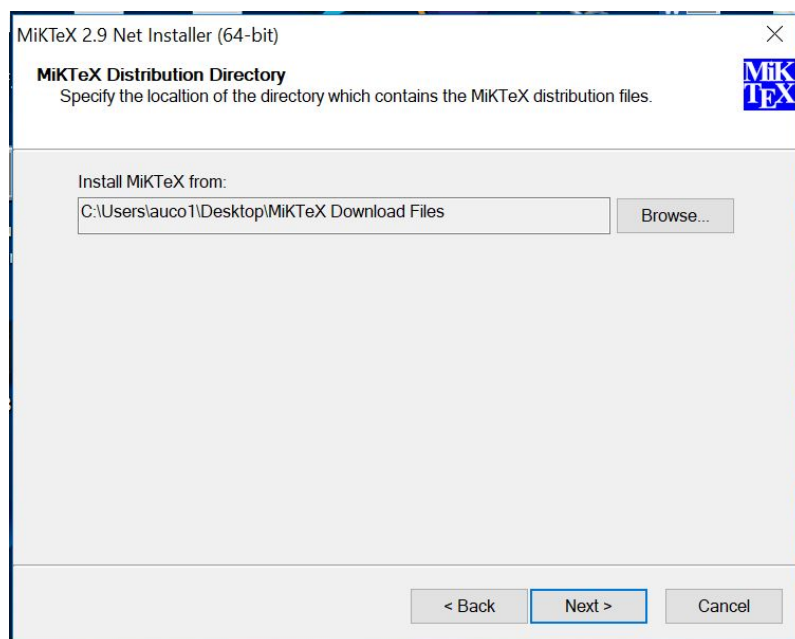
Sau đó, chọn quyền truy cập MiKTeX cho phép ai cũng sử dụng được (Anyone...) hoặc chỉ tài khoản hiện tại (Only for:...).



Hình 2.10: Chọn quyền truy cập MiKTeX

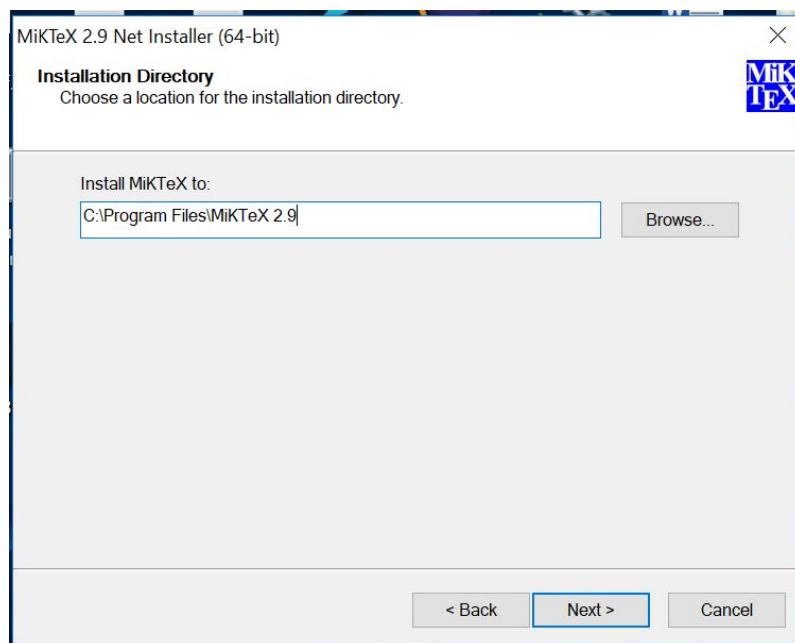


- *Bước 8:* Tiếp theo, trỏ đường dẫn tới thư mục mà ta đã tải các package về.



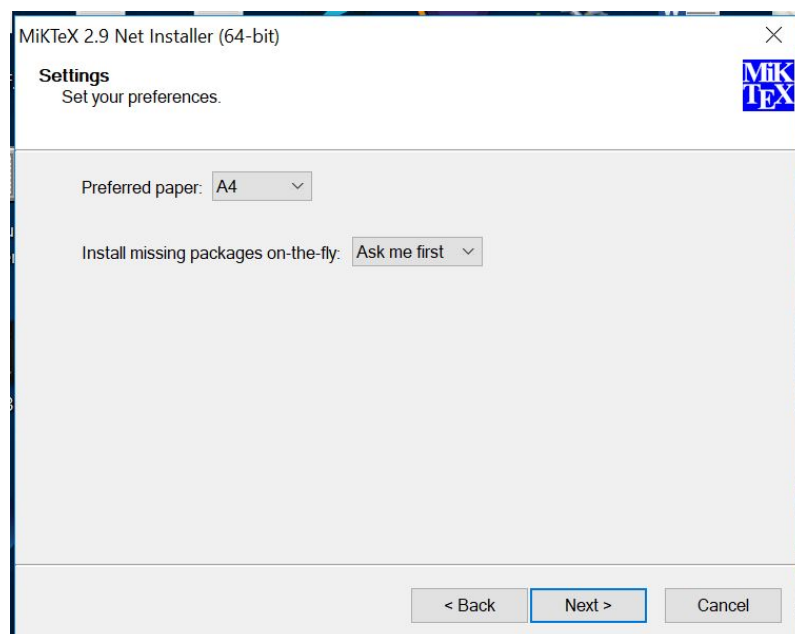
Hình 2.11: Trỏ đường dẫn tới thư mục chứa các package

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



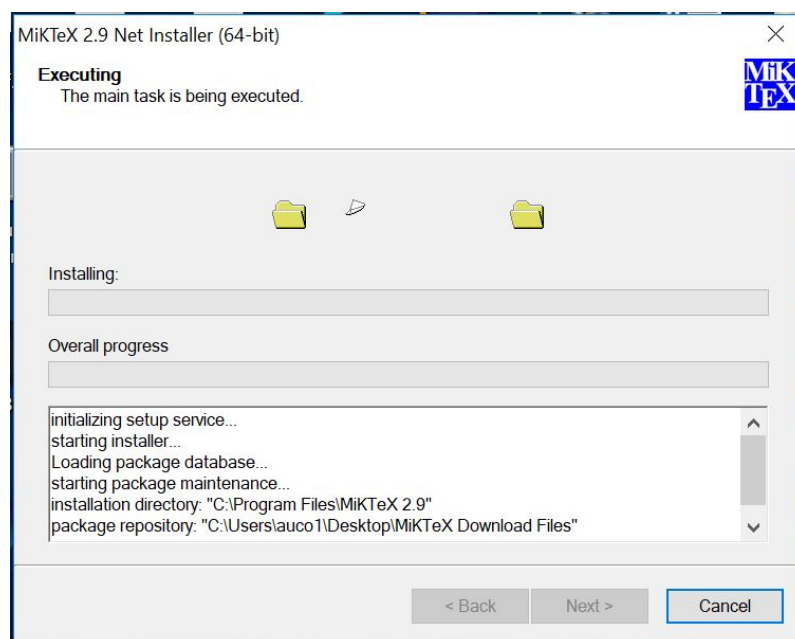
Hình 2.12: Chọn thư mục cài đặt

- *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 package 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 editor LaTeX hỗ trợ Windows, ngoài các editor bản quyền, ta cũng có các trình mã nguồn mở, miễn phí. Trong các editor LaTeX mã nguồn mở cho Windows, TeXstudio được sử dụng thường xuyên nhất, người dùng quan tâm có thể tham khảo và tải editor này tại <https://www.texstudio.org/>.

## 2.2 Hướng dẫn sơ lược

### 2.2.1 Đặc điểm của L<sup>A</sup>T<sub>E</sub>X

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 file chứa nội dung và các “mô tả” đó bằng các editor hỗ trợ, các file đó có tên mở rộng (extension) là `.tex` hay còn được gọi là LaTeX input file. Các file input sau đó sẽ được biên dịch ra code TeX bởi các editor, sử dụng các macro và định nghĩa có trong distribution và xây dựng (build) thành file 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!

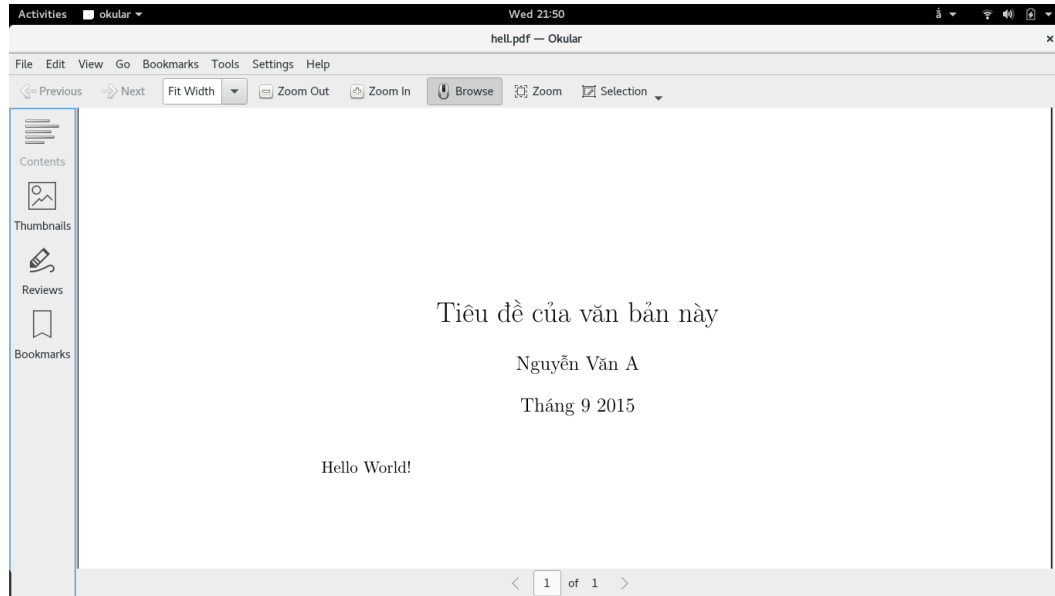
Với các trình xử lý văn bản **WYSIWYG**, trước tiên ta sẽ chọn font, cỡ chữ,... cũng như các hiệu ứng khác nhau nhằm gợi ý người đọc ý nghĩa của nội dung (ví dụ tên người thì in nghiêng, tiêu đề cỡ chữ to, in đậm, căng lề giữa,...). Đối với LaTeX, ta sẽ trình bày trong file input như sau:

```
\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 package `inputenc` với option **utf8** để xử lý các kí tự utf-8
- Sử dụng package `babel` với option `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 file class `article`, class là một file package 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 package `babel` và `inputenc` cách định dạng và mã hoá tiếng Việt. Sau khi build file `.tex` trên ta được văn bản:



**Hình 2.15:** Kết quả sau khi build file input

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?

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 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 package <sup>1</sup> cho phép người dùng tạo 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:

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

<sup>1</sup>Một số package không có sẵn trong distribution và phải được tải bổ sung

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 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 file input. 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 build file 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 editor 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 [14]. Người dùng có thể vào trang hướng dẫn của ShareLaTeX <sup>2</sup>, các diễn đàn như TeX Stack Exchange <sup>3</sup> 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.

<sup>2</sup><https://www.sharelatex.com/learn>

<sup>3</sup><https://tex.stackexchange.com/>

## Cấu trúc cơ bản của một file $\text{\LaTeX}$

Điều đầu tiên ta cần làm là tạo một file `.tex`, sử dụng bất kì trình soạn thảo nào (ưu tiên sử dụng các editor  $\text{\LaTeX}$  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:

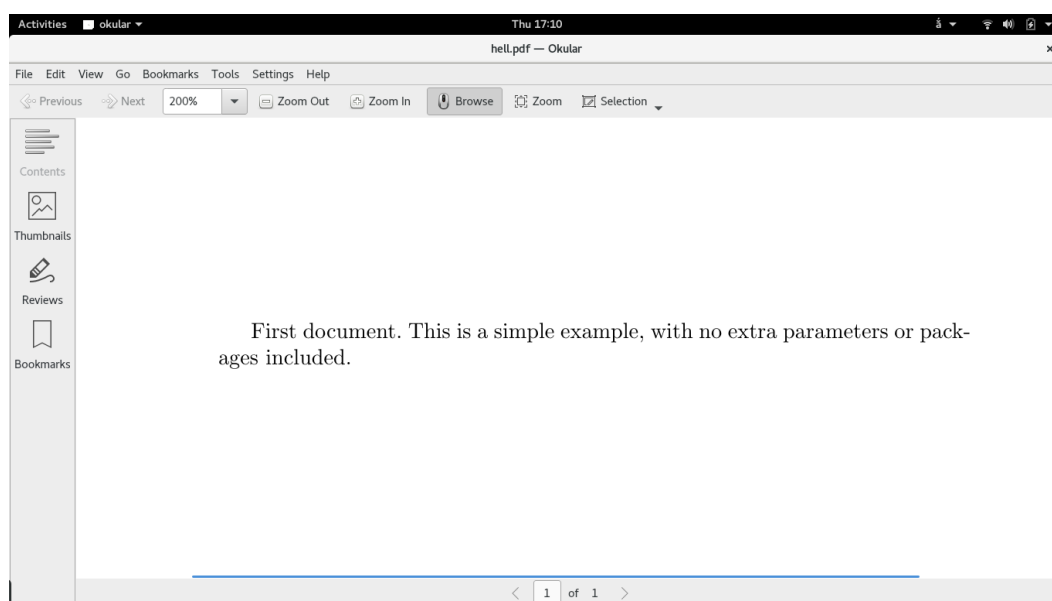
```
\documentclass{article}
```

```
\begin{document}
```

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

```
\end{document}
```

Và kết quả.



Hình 2.16: Ví dụ một văn bản  $\text{\LaTeX}$  đơn giản

Ta có thể thấy  $\text{\LaTeX}$  tự động canh lề và thụt đầu dòng cho đoạn văn. Khi viết file input, điều đầu tiên ta cần làm là khai báo class của văn bản với câu lệnh `\documentclass[option]{class}`, trong đó, **option** đóng vai trò như thông số của class có thể được bỏ trống, mỗi class sẽ có các option khác nhau, và **class** là tên của file class mà ta muốn sử dụng. Thông thường, tên class sẽ ứng với loại văn bản mà người dùng muốn soạn thảo như `article` hoặc `book`, `report`,... Mỗi class 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 tag `\begin{document}` và `\end{document}`, bất kì câu lệnh hay nội dung nào được đánh vào giữa hai tag 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 file LaTeX, phần trước tag `\begin{document}` được gọi là *tiền tố* (*preamble*). Ở phần tiền tố này, ta tiến hành các khai báo class, package và sử dụng các câu lệnh cung cấp thông số dưới dạng *key-value* hoặc nội dung cho package, class (nếu có).

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

Ở đây class `article` đã được khai báo thêm hai option, đó là **a4paper** và **12pt**. Các option 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. Class `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 option và thông tin của class này và tất cả các class cơ bản của LaTeX đều nằm trong tài liệu hướng dẫn [6].

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

Hai câu lệnh `\usepackage[option]{package}` cũng tương tự như `\documentclass` với hai đầu vào **option** là thông số thêm không bắt buộc và **package** là tên package



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 nó, tuy nhiên, khai báo package vẫn rất cần thiết vì sẽ có trường hợp (nếu không muốn nói là thường xuyên) ta cần các package bên ngoài bổ sung thêm nhiều chức năng cho LaTeX để hoàn thành các văn bản phức tạp.

Ví dụ trên sử dụng package `inputenc`, đây là package chứa các bộ mã hoá (encoding) các kí tự văn bản, option `utf8` báo cho package 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 package này thường hay xuất hiện trong các văn bản. Package tiếp theo là gói hỗ trợ đa ngôn ngữ `babel`. Khi văn bản có xuất hiện ngôn ngữ không phải tiếng anh, `babel` kết hợp với `inputenc` để xử lý ngôn ngữ đó, option của package này phần nhiều là tên các ngôn ngữ được xây dựng định nghĩa trong gói.

Mọi chi tiết của hai package trên người dùng có thể tham khảo trong tài liệu hướng dẫn của từng gói `inputenc` [7], `babel` [10].

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

Ba câu lệnh này cung cấp nội dung định dạng cho class `article`, chính vì thế chúng chỉ nên được đặt ở tiền tố, dựa trên định nghĩa của từng câu lệnh này trong class mà LaTeX sẽ tiến hành định dạng mẫu nội dung được “markup”. Đúng như tên gọi, `\title{text}` dùng để cung cấp cho class 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 output cho tới khi người dùng nhập lệnh `\maketitle` nằm trong `\begin{document}`. Khi bắt gặp câu lệnh đó LaTeX sẽ định dạng các thông tin được cung cấp trong các tag ở tiền tố và in ra kết quả.

### Các định dạng cơ bản

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

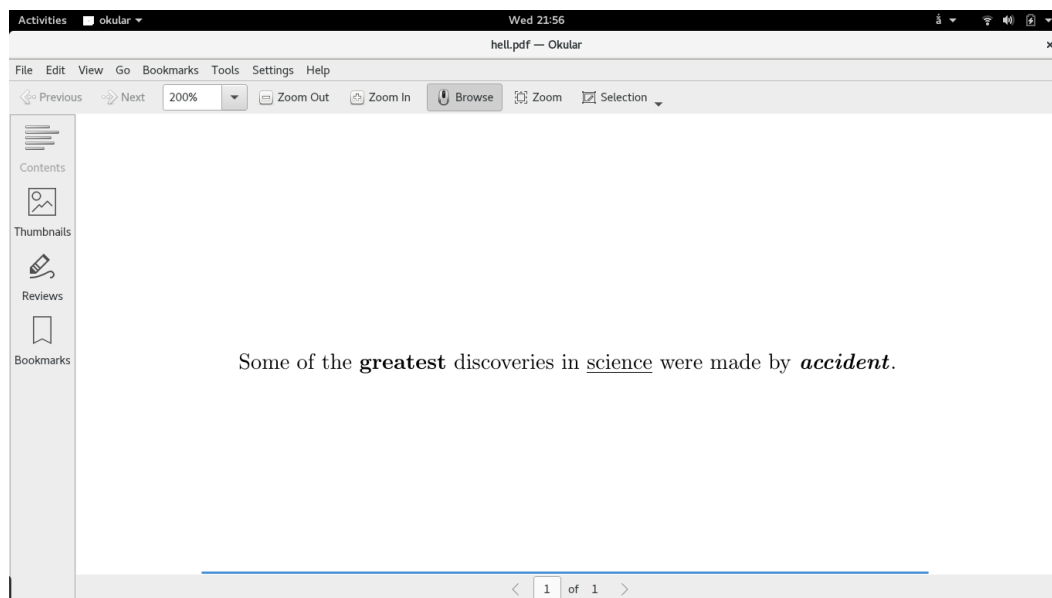
- **In đậm (Bold)**: Nhập đoạn cần in đậm vào câu lệnh `\textbf{text}` (viết tắt của text bold font).
- *In nghiêng (Italics)*: Ta sử dụng câu lệnh `\textit{text}` để đánh dấu đoạn văn bản cần in nghiêng.

- 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:



**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:

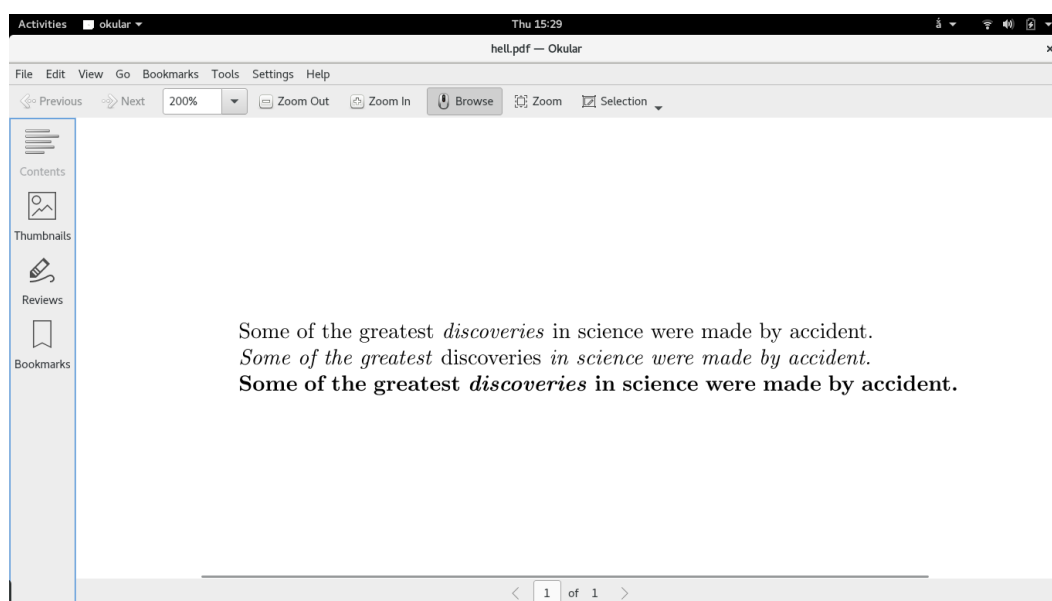
```

\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 file input có xuống dòng (linebreak) ở nhiều nơi, nhưng văn bản kết quả lại chỉ là một dòng duy nhất. Thông thường LaTeX sẽ mặc kệ linebreak trong văn bản nội dung, nếu ta cần phải xuống hàng ở một đoạn cụ thể nào đó ta phải nhấn ENTER hai lần (xuống hàng hai lần) tại đoạn đó như ví dụ hình 2.18 hoặc sử dụng `\\` hay `\par` tại nơi ta muốn xuống hàng như ví dụ sau.

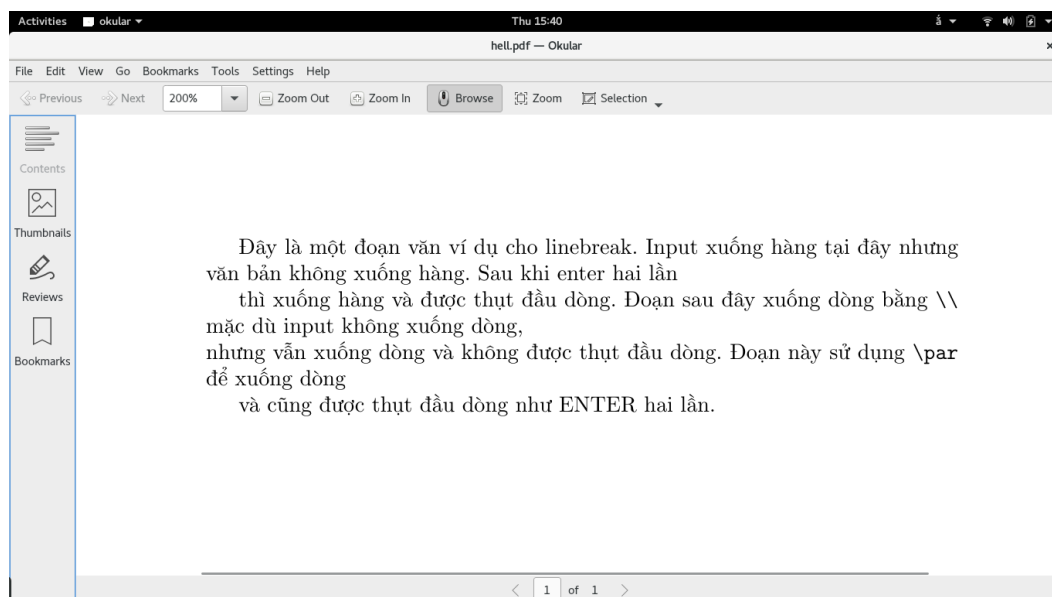
```
\begin{document}
```

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 linebreak và tiếp tục in thẳng cho tới khi hết chiều dài cho phép của một dòng. ENTER hai lần sẽ tương đương với câu lệnh `\par`, ngoài xuống dòng còn tiến hành canh chỉnh cho đoạn văn tiếp theo, còn `\\` chỉ đơn thuần là bổ sung linebreak. Ta thấy trong đoạn văn ví dụ, 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:

```
\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}
```

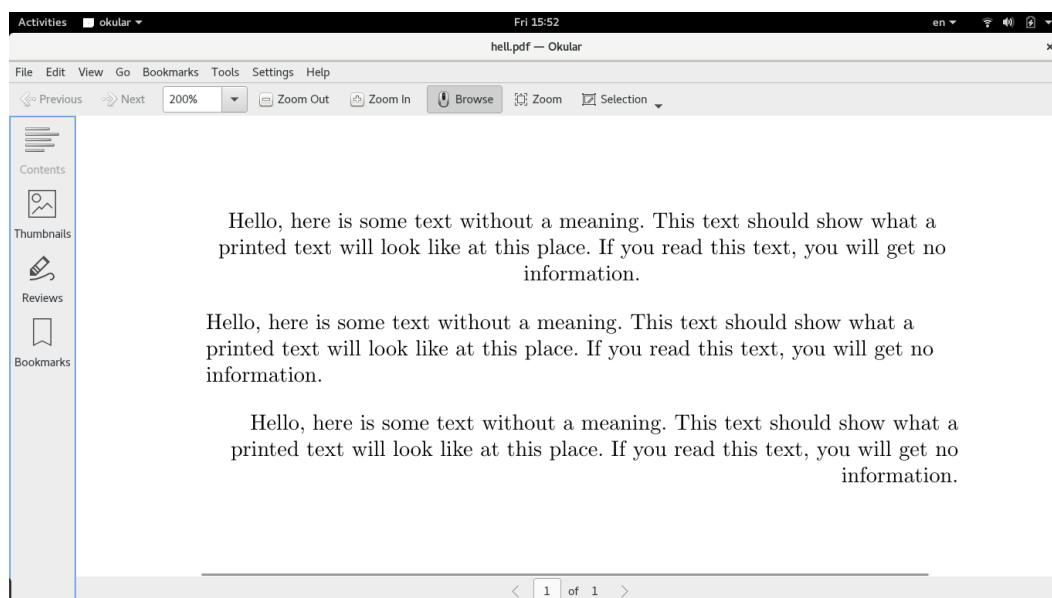
Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information.

```
\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

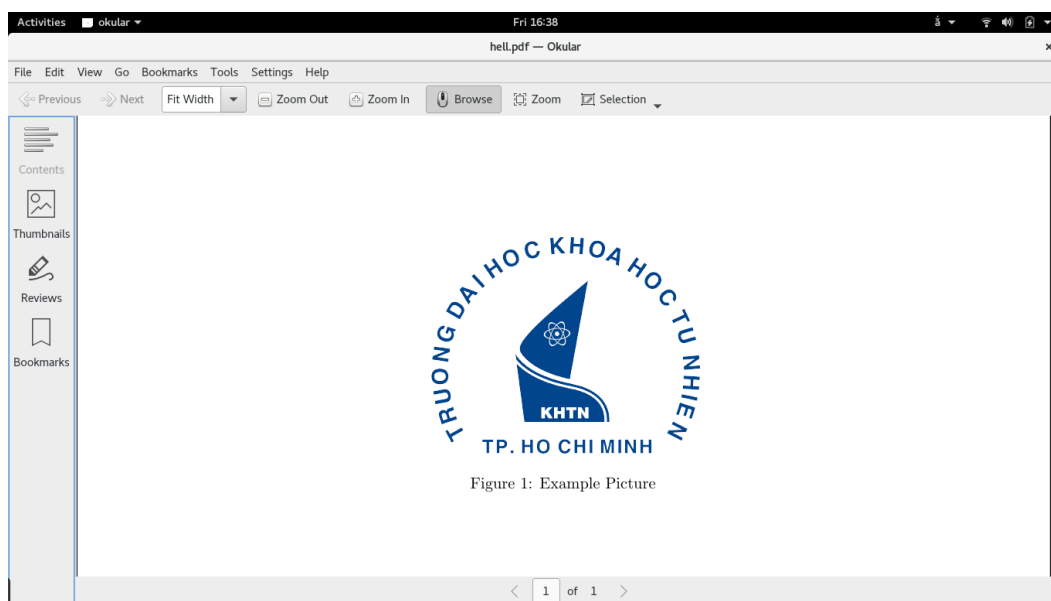
Như có thể thấy, ta chỉ cần đánh đoạn văn bản cần căn chỉnh vào giữa hai tag `\begin{env}...``\end{env}`, trong đó *env* là tên của môi trường (environment) canh chỉnh: ***center***, ***flushleft*** hoặc ***flushright***.

## 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}
```



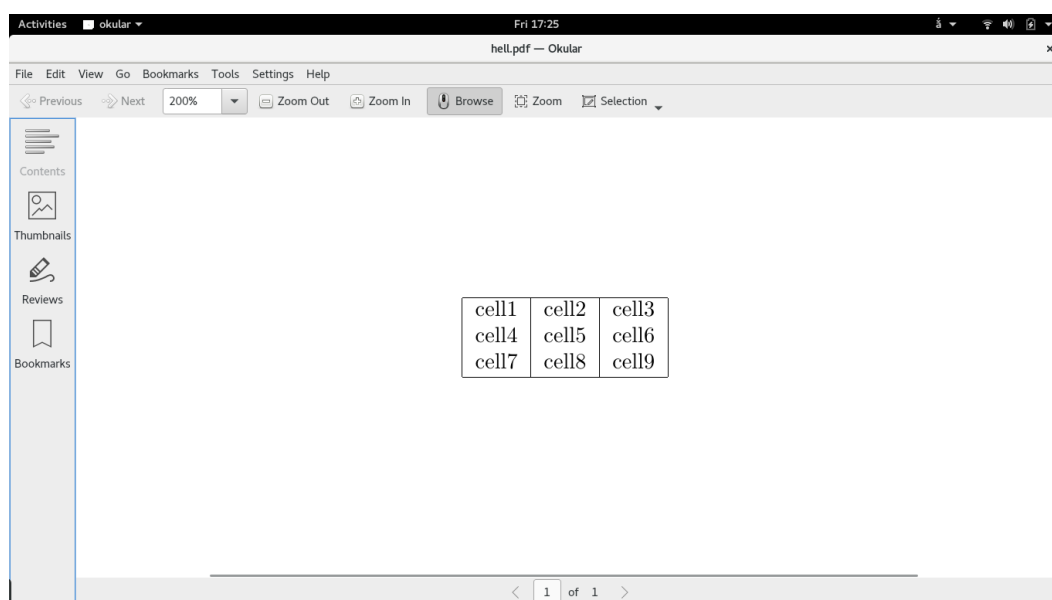
**Hình 2.21:** Ví dụ thêm hình trong văn bản

LaTeX cần phải tải (load) package `graphicx` để thêm hình ảnh, đây là một trong số các package cơ bản nên sẽ kèm theo trong distribution của LaTeX. Để sử dụng package này ta chỉ việc khai báo `\usepackage{graphicx}`, tiếp theo là khai báo đường dẫn chứa file 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, ở ví

dụ sử dụng `h` là đặt hình tại vị trí đặt câu lệnh (so với các đối tượng khác trong file input). `\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 key-value, 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 đánh 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, `{ |c|c|c| }` ý nói 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 `|` 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 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...

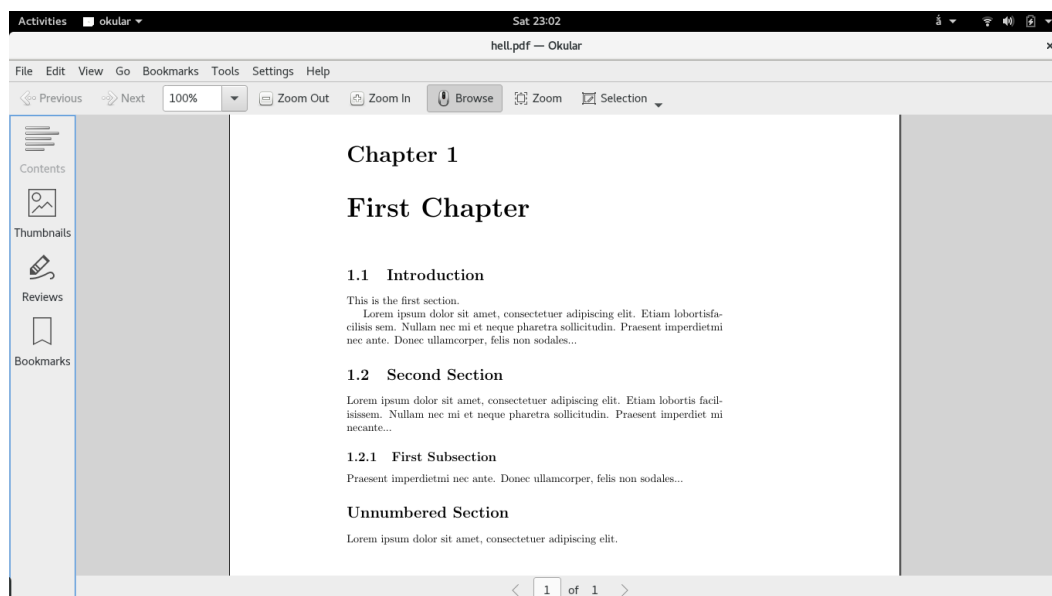
```
\subsection{First Subsection}
```

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 trên 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 class `report` và `book`.

---

## CHƯƠNG 3

---

# Thiết kế định dạng văn bản riêng trong L<sup>A</sup>T<sub>E</sub>X

### 3.1 Tổng quan về package và class

Như đã nói ở phần 1.3, người dùng hoàn toàn có thể tạo cho mình một file class hoặc package 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 package và class được tác giả chia sẻ cho cộng đồng (La)TeX sẽ được bảo hộ bằng giấy phép LPPL<sup>1</sup> và được lưu trữ trong Comprehensive TeX Archive Network (CTAN), là tập hợp các trang web dùng để lưu trữ các package, class 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 giới<sup>2</sup>, danh sách các nguồn tải ở hình 2.5 là một trong số đó. Do được kết nối với các LaTeX distribution, nên người dùng có thể dễ dàng tải các package trên CTAN thông qua chúng, giúp tác giả package, class có thể dễ dàng phân phối, bảo trì và cập nhật sản phẩm của mình.

Người dùng có thể dễ dàng viết package và class bằng bất cứ trình soạn thảo nào và chỉ cần đặt tên mở rộng của file thành .cls (class) hoặc .sty (package), tuy nhiên, ta vẫn nên ưu tiên sử dụng trình soạn thảo hỗ trợ LaTeX với lí 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 class hay package. Thông thường, nếu muốn xây dựng, thiết kế bố cục, cấu trúc logic của một loại văn bản tùy ý nào đó, ta sẽ viết class, 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 viết package.

---

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

<sup>2</sup><https://www.ctan.org/mirrors/>

## 3.2 Giới thiệu class đề tài

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 file class LaTeX, với các câu lệnh, canh chỉnh theo đúng quy định về bố cục luận văn, bài báo cáo của trường và bộ môn. File class sản phẩm dựa trên class `book` của LaTeX, mọi chi tiết về class `book` xin hãy tham khảo tài liệu thông tin các class tiêu chuẩn của LaTeX [6]. Ý tưởng một số câu lệnh trong đề tài này dựa trên class `gsemthesis`, là mẫu (template) đề tài tiến sĩ của trường đại học Geneva, Thụy Sĩ, thông tin chi tiết về class này được nêu trong tài liệu hướng dẫn [8].

## 3.3 Cấu trúc của class

Phần này chủ yếu tập trung vào cấu trúc bên trong của file class với các câu lệnh, khai báo, thông số cũng như các package 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{myclass}
```

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 $\epsilon$  mà người viết file 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 distribution tự động tải về bản LaTeX mới nhất, người dùng không cần quá quan tâm đến thông số này.

Câu `\ProvidesClass{myclass}` dùng để khai báo tên của file class đang viết và phải trùng với tên của file `.cls` được lưu, ở đây, tên khai báo là “`myclass`” và tên của file được lưu là `myclass.cls`.

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

Là câu lệnh dùng để load file class mà sản phẩm ta viết sẽ dựa trên, như đã nói ở 3.2, class này dựa trên `book` của LaTeX, kèm theo các option 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}
```

Class sử dụng package `inputenc`, option **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 load package `etoolbox` chứa các câu lệnh hỗ trợ viết các file class, package cho LaTeX [11].

```
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{myclass}{Unknown option '\
    CurrentOption'}}
19 \ProcessOptions\relax
```

Các câu lệnh `\DeclareOption{option name}{code}` trên dùng để khai báo các option có trong class này mà người dùng có thể khai báo khi gọi class bằng câu lệnh `\documentclass[option]{class}`, với *code*, dùng để định nghĩa option, LaTeX sẽ tiến hành các câu lệnh trong phần này nếu người dùng khai báo *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 option có trong class, nó có chức năng báo cho LaTeX phải xử lý các option mà người dùng khai báo (nếu có) trước khi tiến hành các câu lệnh tiếp theo.

Các option mà class này có được định nghĩa thông qua các câu lệnh trên như sau:

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

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

Câu lệnh `\DeclareOption{twoside}{\OptionNotUsed}` dùng để ra cảnh báo nếu người dùng sử dụng option in hai mặt **twoside**, do chuẩn bài báo cáo chỉ yêu cầu in một mặt, người dùng có thể mở file class này và thay đổi câu lệnh thành như trong trường hợp khổ giấy A5 trên, sử dụng `\PassOptionsToClass{\CurrentOption}{book}`, nếu muốn sử dụng option **twoside**.

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

Class định nghĩa option 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 option **vietnamese**, class 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 phần code dòng 12 và 13, đặt giá trị **true** cho các biến boolean mới mỗi khi người dùng khai báo option.

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

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

### 3.3.2 Khai báo package và thiết lập

Dưới đây là các đoạn lệnh dùng để khai báo và thiết lập thông số cho các package được sử dụng trong file class này. Khi khai báo class đồng nghĩa với việc người dùng đã khai báo các package tích hợp, người dùng hoàn toàn có thể thay đổi các thiết lập sau tùy thích sử dụng câu lệnh tương tự với class khi soạn thảo file input.

```

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` sẽ thực hiện code trong phần *true condition*, và ngược lại, code trong 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 option **vietnamese**, khi đó, câu lệnh thực hiện câu lệnh ở phần *true condition*, tức ba câu lệnh từ dòng 22 tới 24.

Ba câu lệnh ở điều kiện `true`, dùng để load package `babel` option **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`. Package `hyphsubst`, dùng để ngăn LaTeX đưa cảnh báo về ngắt chữ khi xuống dòng (file input vẫn được build bình thường khi có cảnh báo này, việc sử dụng gói `hyphsubst` chỉ đơn thuần nhằm hạn chế các cảnh báo không cần thiết). Package `indentfirst` cho phép LaTeX thụt đầu dòng đoạn văn (paragraph) đầu tiên của một chương, đề mục,...

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

Phần khai báo tiếp theo sẽ là về các package 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}

```

Package **geometry** dùng để canh chỉnh lề cho trang giấy và là một trong những gói cơ bản của LaTeX, có sẵn khi tải distribution, option của package này thường là các thông số, kích thước cho lề của mặt giấy. Class sử dụng package 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.

Package **graphicx** như đã nhắc tới ở phần 2.2.2, dùng để thêm hình ảnh vào văn bản và là một trong những package cơ bản, có sẵn. Package **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 gom nhiều cột dòng người dùng có thể xem hướng dẫn, ví dụ ở ShareLateX [14] hoặc tài liệu của package này [9].

Package **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 [14].

Package **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 package này và môi trường **tabular** sẽ được tìm thấy trong tài liệu hướng dẫn [12].

Package **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 package `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 package này được chính **AMS** phát triển.

Package `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], package `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 “comment” trong một file code). Package `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 package `adjustbox`, `multirow`, `wrapfig`, `tocbibind`, `notoccite`, `path` có thể không có sẵn trong distribution, người dùng cần phải tải bổ sung nếu hệ thống báo thiếu, cách tải package LaTeX bổ sung trong Linux được nêu ở phần 2.1.1.

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

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

Tương tự như gói `path`, package `url` cung cấp câu lệnh `\url| |` để ta đánh dấu địa chỉ URL của website, option `obeyspaces` cho phép hiển thị khoảng cách (nếu có) trong địa chỉ (điều hiếm khi xảy ra), package 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 [5] của package.



# 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] Donald Arseneau. *url.sty version 3.4*. Sept. 16, 2013. URL: <http://mirror.kku.ac.th/CTAN/macros/latex/contrib/url/url.pdf>.
- [6] 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>.
- [7] Alan Jeffrey and Frank Mittelbach. *inputenc.sty*. v1.2c. Mar. 17, 2015. URL: <http://mirror.kku.ac.th/CTAN/macros/latex/base/inputenc.pdf>.
- [8] 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>.
- [9] 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>.
- [10] 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>.
- [11] 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>.
- [12] 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>.
- [13] *Introduction to LaTeX*. LaTeX3 Project Team. URL: <https://www.latex-project.org/about/>.

- [14] *Learn LaTeX in 30 minutes*. ShareLaTeX team. URL: [https://www.sharelatex.com/learn/Learn\\_LaTeX\\_in\\_30\\_minutes](https://www.sharelatex.com/learn/Learn_LaTeX_in_30_minutes).
- [15] *The Kile Handbook*. Jonathan Pechta et al. URL: <https://docs.kde.org/stable4/en/extragear-office/kile/index.html>.
- [16] *What are TeX and its friends?* CTAN team. URL: <https://www.ctan.org/tex/>.