

CONTENT

[#Introduction](#)

[#Researching methodology](#)

[#Ideas and Projects.](#)

[#analog-digital design.](#)

[#computer architecture](#)

[#micro-controllers](#)

[#Galileo](#)

[#FriendlyARM](#)

[#arduino](#)

[#NoC](#)

[#SoC](#)

[#VIM](#)

[#Linux commands](#)

[#GNUPlot](#)

[#LaTEX skills](#)

[#C&C++](#)

[#SystemC&VHDL](#)

[#Shell scripts](#)

[#Assembly language](#)

[#basic knowledge](#)

[#HOWTO](#)

[#Book and Best Ref.](#)

[#Important conversations](#)

[#vocabulary&usingwords](#)

[#EGFLY](#)

[#English Notes](#)

[#Job Requirement](#)

[#useful links for studying](#)

[#My dinary!](#)

[#Abbreviation](#)

[#Bibliography](#)

Lời giới thiệu:

Ngày nay, việc nghiên cứu khoa học trong lĩnh vực kỹ thuật nói riêng và các lĩnh vực khác nói chung đã ngày càng trở nên dễ dàng với sự phát triển và sự thân thiện của Internet. Việc sử dụng Internet để học hỏi, nghiên cứu ngày càng được nhiều người biết đến. Tuy nhiên, đôi khi những người bắt đầu về một chuyên ngành cụ thể sẽ có thể còn một vài khó khăn trong việc bắt đầu với cái gì? Thông qua tài liệu này, tác giả muốn gửi đến chính bản thân mình (khi đọc lại) cũng như những ai muốn nghiên cứu trong lĩnh vực Hệ thống nhúng (Embedded System) có thể có một cái nhìn tổng quan đến chi tiết về một hướng nghiên cứu!

Hệ thống nhúng là rất rộng, ở đây, tác giả chủ yếu ghi chép lại những gì mình làm, nhằm sau này người đọc có thể dựa theo những nội dung trong phần mục lục có thể tìm đến những phần liên quan để đọc và nghiên cứu tiếp các nội dung đó. Nếu tài liệu có thiếu sót gì hoặc có đóng góp chia sẻ gì, xin được tiếp thu ý kiến qua email riêng của tác giả: dvnam@ictu.edu.vn. Trân trọng cảm ơn!

Một vài chú ý:

#: Đây là ký hiệu cho một vấn đề tác giả muốn đưa ra để trả lời.

!#: Đây là đáp án tự giải quyết bởi tác giả.

?#: Đây là ký hiệu mang ý nghĩa vấn đề này chưa được giải quyết hoặc hiểu triệt để.

Method in research:

Ideas => planning => do it immediately => smart thinking => >>>???

Researching methodology

The all following topics and questions will present about How could we do in our researching job.

Welcome and hope it will be a helpful guide!

#How to read a paper?

!#

>>>

How to choose a topic to research?

!#>>

#Structure of a scientific paper?

!# (I read these topics from How to Write a Paper 2013). p12/167

IMRAD = Introduction+Methods+Results and Discussion.

Introduction: What Question was asked?

Methods: How was it studied?

Results: What was found?

Discussion: What do the finding mean?

#What should know about the Introduction part?

!#

Phần giới thiệu nên tóm tắt làm rõ ràng rằng nghiên cứu của bạn đã giải quyết vấn đề gì? Để dẫn dắt người đọc nắm được điểm chính yếu của nghiên cứu, việc tóm tắt một vài nghiên cứu có liên quan là cần thiết trong phần này.

Trong phần giới thiệu không nhất thiết phải đưa ra review của các bài nghiên cứu trước đây mà chỉ cần dẫn chứng (trích dẫn/cite) chúng để minh chứng cho đề xuất nghiên cứu của bạn.

* Trong phần giới thiệu (Introduction).

Trong phần giới thiệu nên tóm lược ngắn gọn để nói cho người đọc biết tại sao bạn lại thực hiện nghiên cứu này. trước khi bắt đầu ngồi xuống và viết phần giới thiệu, bạn cần trả lời một vài câu hỏi cơ bản:

- Tôi cần nói cái gì?
 - Ý nghĩa của việc nói?
 - Việc nói, diễn đạt đã chuẩn theo định dạng chưa?
 - Thông tin cần truyền đạt đến người nghe là gì?
 - Thông tin cần truyền đạt có phù hợp với tạp chí?
- ?? sẽ chỉnh mấy lưu ý trên sau vì dịch chưa chuẩn lắm.

Tell readers why you have undertaken the study.

The main job of the introduction is to tell readers why you have undertaken the study. If you set out to answer a question that really interested you, then you will have little difficulty. But if your main reason for undertaking the study was to have something to add to your curriculum vitae, it will show.

If your audience is interested in the answer to these questions then they may well be tempted to read the paper and, if you have defined your audience and selected the right journal, they should be interested.

More commonly, you will be building on scientific work already published. It then becomes essential to make clear how your work adds importantly to what has gone before.

Clarify what your work adds.

Editors will not want to publish – and readers will not want to read – studies that simply repeat what has been done several time before. Indeed, you should not be undertaking a study or writing a paper unless you are confident that it adds importantly to what has gone before.

Usually, it is not so easy to make clear how your study is better than previous ones, and this is where the temptation arises to give a detailed critique of everything that has ever gone before. You will be particularly tempted to do this because, if you are serious about your study, you will have spent hours finding and reading all the relevant literature. The very best introductions will include a systematic review of all the work that has gone before and a demonstration that new work is needed.

The move towards systematic reviews is one of the most important developments in science and scientific writing in the past 20 years. We now understand that most reviews are highly selective in the evidence they adduce and often wrong in the conclusions they reach. When undertaking a systematic review, an author poses a clear question, gathers all relevant information (published in whatever language or unpublished), discards the scientifically weak material, synthesizes the remaining information and then draws a conclusion.

To undertake such a review is clearly a major task, but this ideally is what you should do before you begin a new study. You should then undertake the study only if the question cannot be answered and if your study will contribute importantly to producing an answer. You should include a brief account of the review in the introduction. Readers will then fully understand how your study fits with what has gone before and why it is important.

The one page begins with questions that even when writing for other journals it will be useful for you to answer in one or at most two sentences: What is the study question? What is the summary answer? What is known and what does your paper add?

Make sure that you are aware of earlier studies

I've already emphasized the importance of locating earlier studies. Before beginning a study, authors should seek the help of librarians in finding any earlier studies. Authors should also make personal contact with people who are experts in the subject and who may know of published studies that library searches do not find, unpublished studies or studies currently under way. It's also a good idea to find the latest possible review on the subject and search the references and to look at the abstracts of meetings on the subject.

Editors increasingly want to see evidence that authors have worked hard to make sure that they know of studies directly related to theirs. This is particularly important when editor's first reaction to a paper is “Surely we know this already”.

Don't baffle your readers.

Give the study's design but not the conclusion.

Conclusions

To write an effective introduction you must know your audience, keep it short, tell readers why you have done the study and explain why it's important, convince them that it is better than what has gone before and try as hard as you can to hook them in the first line.

#What should know about the Methods part?

!#

Bạn cần mô tả, về thứ tự logic, cách thiết kế và thực thi nghiên cứu của bạn. và cả cách thức bạn phân tích, xử lý số liệu.

Khi người đọc tìm đến phần phương pháp (methods) thì họ muốn tìm kiếm nhiều hơn những gì đề tài nghiên cứu đạt được. Phần phương thức cần trả lời các câu hỏi “ai, cái gì? tại sao? khi nào? và ở đâu?”. Thậm chí nó nên khẳng định rằng các giả thuyết đã được kiểm thử.

thống kê:

đưa ra chính xác các bài kiểm thử đã được sử dụng để phân tích số liệu, và cần trích dẫn thêm các nguồn tham khảo phù hợp nếu như phương pháp test đó là chưa phổ biến. Chỉ rõ phần mềm, phiên bản bạn sử dụng, cũng như mô tả rõ ràng các giả thiết và các số liệu dùng để đánh giá các vấn đề mà bạn chọn như các phân bố cơ bản... Các bài kiểm thử thống kê phụ thuộc vào các giả thiết. Đôi khi thì sự phân bố dữ liệu không rõ ràng trước khi kết thúc bài nghiên cứu, do vậy mà việc lựa chọn các kiểu test là cần cẩn trọng...???

Thiết kế:

*** Trong phần phương thức thực hiện/phương pháp nghiên cứu (Methods). (ref from p26/167)**

You must describe, in logical sequence, how your study was designed and executed, and how you analyzed the data. Proper planning detects mistakes before they happen. Write the methods section, in full, before you start the study. Ask an experienced colleague to look it over. The challenge of setting down what you intend to do is also a very useful exercise – far better than discovering predictable flaws after months of hard work.

Testing hypotheses

>>>>>>>>???

#What should know about the Results part?

!#

*** Trong phần kết quả (Results). (ref from p34/167)**

how to write this part? (results part), some notes for writing.

!#

Phần kết quả là phần báo cáo về kết quả điều tra của dự án/đề tài. Phần này gồm văn bản, các bảng số liệu và hình ảnh. Thông thường thì phần này yêu cầu độ chính xác cao, tránh giải thích hay báo cáo các số liệu cần thiết để chứng minh sự không đúng của cơ sở lý thuyết. Hơn nữa, phần này cũng cần phải theo trật tự logic, thông thường thứ tự logic là theo trình tự của phương pháp thực hiện (methods).

Trước khi viết phần này, cần tổ chức các bảng, hình ảnh theo thứ tự trình bày của chúng. cuối cùng, cần kiểm tra lại theo sự hướng dẫn của thầy về định dạng, chiều dài, số lượng hình ảnh cũng như bảng số liệu sao cho phù hợp với tạp chí định submit.

TEXT:

Phần này nên mô tả đầy đủ đối tượng nghiên cứu. Các kết quả nghiên cứu cần được đánh giá, phân tích một cách chính xác, rõ ràng, bởi người đọc sẽ cần biết được đầy đủ cách thức thu thập dữ liệu. Chú ý không lặp lại thông tin đã được cung cấp trong các bảng số liệu. và các kết quả thường không bao giờ có kèm theo tham khảo (references) và chú ý tránh sử dụng các từ viết tắt quá nhiều vì việc lạm dụng quá đáng từ viết tắt có thể dẫn đến việc vướng víu cho người đọc.

Với mỗi một đoạn dùng một câu chủ đề thể hiện được thông tin về bộ dữ liệu đã được tiết lộ. Điều này có thể được tóm tắt sau khi đưa ra hình vẽ/bảng số liệu về kết quả được tìm ra. Câu chủ đề của đoạn (topic sentence) không nên quá đơn giản chỉ là lặp lại tiêu đề của hình vẽ hay bảng số liệu. mà nó nên là câu trả lời cho những lý thuyết nghiên cứu/hướng nghiên cứu đã được đề cập trước đó (có thể trong bài báo/tài liệu tham khảo). Sự thay đổi chênh lệch của các kết quả cần nằm trong khoảng chuẩn chênh lệch hay tiêu chuẩn lỗi. hay có tối thiểu 95% là tin tưởng chính xác. Trong một vài trường hợp thì số liệu được cung cấp là số liệu thô.

Các số liệu đã được cung cấp qua bảng số liệu và hình vẽ thì không nên trình bày lại bằng văn bản (text). Có thể trình bày mô tả bảng/hình ảnh của kết quả theo như dạng 1 trong kỳ thi IELTS, tức là đưa ra những điểm mấu chốt, cốt lõi, điểm nhấn mạnh trong kết quả và trong hình vẽ thể hiện sự thay đổi lớn đáng chú ý (>>?? nên tích lũy thêm kinh nghiệm để viết lại chỗ này)

Kiểm tra cẩn thận lại tính nhất quán trong quá trình viết bài báo. (Từ đầu đến cuối của bài báo nên nhất quán sử dụng các định dạng cũng như các thuật ngữ đã sử dụng).

Thường sử dụng thì quá khứ đơn trong phần này. khi các thí nghiệm đã hoàn thành. Mọi liên hệ dữ liệu trong phần này tránh sử dụng các thuật ngữ chuyên ngành quá sâu chứa hàm ý hay các cơ chế hoạt động. Cũng nên tránh các cụm từ định tính (??? chỗ này cần đọc kỹ thêm và trình bày sau! Trang 34/167 trước mục table).

BẢNG (TABLES)

Các bảng nên cung cấp thông tin tường thuật về các kết quả nghiên cứu với cách thức đơn giản, dễ nhìn để người đọc tiện theo dõi. Một vài tạp chí có thể cho phép sử dụng màu để làm các dữ liệu trong bảng trở nên nổi bật và dễ quan sát cũng vì mục đích này. Mỗi bảng nên để riêng ở một trang và có đánh số thứ tự để dễ dàng tìm kiếm đến đoạn văn bản liên quan. Kiểm tra cấu trúc, yêu cầu trình bày, đánh dấu bảng của mỗi một tạp chí. Các bảng nên thiết kế nhỏ nhất có thể để giải thích phần giả thuyết. Đừng nên lặp lại các bảng trong các bảng khác hay hình ảnh khác. Nếu sử dụng các bảng khác hay các kết quả từ các tài liệu tham khảo khác thì cần ghi chú rõ ràng và cần được sự đồng ý của các tác giả trước.

...

Người đọc không cần phải đối chiếu, tham khảo đến các đoạn text mà vẫn có thể hiểu được thông tin khi nhìn vào các bảng. Mỗi từ viết tắt cần được định nghĩa trong mỗi bảng hoặc cần được ghi chú rõ ràng. Tránh lạm dụng việc viết tắt thái quá, đặc biệt với các từ viết tắt không thường gặp phổ dụng. Sẽ rất vướng víu và không tiện cho người đọc nếu cứ phải lật lại những trang ghi chú từ viết tắt để hiểu được bạn đang trình bày cái gì trong bảng.

CÁC HÌNH VẼ (Figures)

Các hình vẽ dưới dạng lưu đồ, tranh ảnh, đồ thị và video được sử dụng để minh họa các kết quả một cách rõ ràng và sinh động. Khi bạn chuẩn bị các hình ảnh, cần lưu ý rằng chúng sẽ bị thu nhỏ về kích thước khi xuất bản. Việc dùng các hình ảnh nên thuận lợi cho người đọc có thể hiểu được các kết quả. Vì vậy cần chắc chắn rằng tất cả các nhãn đều cần phải chính xác và tránh sử dụng kiểu trang trí không cần thiết.

Phần kết quả của một bài báo có thể là phần quan trọng nhất và dễ dàng viết nhất?? Có thể so sánh việc viết báo với việc viết một câu chuyện ngắn. Sử dụng phương pháp loại suy, rồi bóc trần các sự huyền bí. Phần giới thiệu (introduction) và phần phương pháp thực hiện (methods) cung cấp sự thiết lập khởi tạo các nhân vật trong phim hay của câu chuyện. và phần kết luận (epilogue) và phần thảo luận là tóm lược và móc nối tất cả lại với nhau. Trong phần kết quả là phần mà người đọc mong đợi nhất. Nó sẽ trả lời cho câu hỏi ai hoàn thiện nó? Sự kết hợp khéo léo trong việc sử dụng text, hình ảnh và bảng cung cấp cho người đọc các điểm cần thiết của nghiên cứu (câu chuyện).

#What should know about the Discussion part?

!#

all in one figure.

You can see the following figure to remember the main parts (structure of a discussion part)

should be included them.

Box 5.1 Discussion: overall format

- Statement of principal finding(s)
- Appraisal of methods
- Comparison with previous work
- Clinical and scientific implications (if any)
- Further work
- Conclusion (optional)
- Acknowledgements

You can read more in the book (page 30 (40 of 167)) to know in detail.

- Những phát hiện chính: Dùng hai đến ba câu để tóm lược kết quả nghiên cứu là một cách tốt để bắt đầu viết phần này. Những câu này nên thật rõ ràng, dễ hiểu. Và chúng đôi khi có thể được sử dụng trong phần tóm tắt (abstract). Việc phân tích quá sâu hơn vào số liệu có thể không nên thực hiện trong phần thảo luận này. Nếu bạn bỏ sót một phần phát hiện mới nào đó quan trọng của kết quả, bạn có thể sẽ phải viết lại phần này để chỉ rõ ràng hơn về kết quả đó.
- phương thức thực hiện (methodology): Phần này không giống với phần phương pháp thực hiện (method) trình bày trong phần trước của nghiên cứu. Một vấn đề chung đó là kích thước mẫu, các công thức được mô tả trong phần phương pháp (methods) có thể được optimistic (lạc quan hóa). không có gì phải giấu diếm trong phần này giữa người biên dịch và người định giá (assessor).
- >>> should re-read again and again...

How to write a paper?

!#>>>>

Một tài liệu được bản thân đánh giá là rất tuyệt vời khi bắt đầu làm nghiên cứu hay viết báo khoa học đó chính là tài liệu của tác giả abc với tựa đề abc và được tái bản lần thứ 6 năm 2015.

Tác giả up tài liệu và chia sẻ qua googledriver cá nhân theo link:

<https://drive.google.com/a/ictu.edu.vn/file/d/0B8DOEBogUa-2eEZTRIjWWHNHRjQ/view?usp=sharing>

Mọi thắc mắc xin gửi về email dvnam@ictu.edu.vn

Còn cập nhật!
thân mến!

#How to read a book?

!#>>>>

#Structure of a papers?

#The ideas and Projects (from idea to implement)

#Cần luôn luôn nhớ!

(Khi dậy em Danh học!)

- Một tuần có 7 ngày, anh chỉ về có một đến hơn một ngày, nếu em không tự cố gắng học những ngày còn lại thì anh có là thánh tướng cũng không thể giúp em học tốt hơn được.
- Luôn cố gắng học, tập trung suy nghĩ, thông minh hơn, vận dụng não bộ tốt hơn...
- Sẽ mất khoảng 2 năm để học cách nói, nhưng bạn có lẽ phải mất đến 20 năm để biết được khi nào nên im lặng?!
-

Why do I create this part?

For the following reasons:

- first, I want to improve my own skills in researching via doing projects.
- Second, I will create as much as related projects possible to help my colleagues, students in researching and for myself to get more and more opportunities in finding more money.
- Finally, doing for my passion.

FORM of an project (update on Friday 4th March 2016 at 12:14)

I1: (project 1):

title:

Introduction:

expected results:

status:

>>>

I*: project: for my university ICTU 2016. (first and last time doing like that!)

Title: *NOXIM: A platform of Network-on-Chip Simulating for researching in a testing and verification system.*

Introduction: Hệ thống mạng trên chip những năm gần đây nhận được sự quan tâm lớn của các nhà khoa học nghiên cứu trong lĩnh vực thiết kế chip trên toàn thế giới. Nguyên nhân chính của sự nổi lên như một vấn đề được ưu tiên nghiên cứu đó chính là hệ thống mạng trên chip có thể giải quyết được vấn đề truyền thông phức tạp trong một hệ thống trên chip phức tạp (SoC complexity). Mạng trên chip (Network-on-Chip: NoC) được coi là một hệ thống truyền thông trên chip mở rộng của một hệ thống trên chip (System On Chip: SoC) do bởi tính khả dụng trong truyền thông, việc truyền thông theo kiểu chia sẻ dữ liệu bus dùng chung (shared bus) đã không còn phù hợp trong xu thế các hệ thống trên chip ngày nay tích hợp ngày càng nhiều lõi IP (Intellectual Property) trên cùng một đế chip bán dẫn. // Trong lĩnh vực nghiên cứu về hệ thống mạng trên chip (Network-on-Chip: NoC) thì nổi lên những vấn đề chính như là: thiết kế bộ định tuyến cho mạng (routers), thiết kế bộ giao tiếp mạng (NI: network interface), thiết kế thuật toán định tuyến cho mạng (routing algorithm), etc. Và tất cả một hệ thống truyền thông trên chip sau khi được hình thành thì cần có một cơ chế để kiểm chứng và đánh giá. NOXIM được biết đến là một bộ công cụ mô phỏng hệ thống mạng trên chip chạy trên nền Linux với mã nguồn mở là ngôn ngữ C++ (SytemC). Qua việc kiểm chứng bằng mô phỏng hệ thống NoC với NOXIM, ta có thể có được cái nhìn tổng quan nhất về một hệ thống NoC từ thiết kế đến các thành phần bên trong của hệ thống và các thông số quan trọng cần quan tâm. // Tác giả thực hiện việc tái cấu hình hệ thống NoC với bộ công cụ NOXIM và đưa ra các kịch bản mô phỏng nhằm so sánh các giải thuật định tuyến mạng trên chip khác nhau. // Đề tài được đề xuất và duyệt bởi ĐH CNTT&TT với số

mã NC và...xxx. Các tác giả xin chân thành cảm ơn sự giúp đỡ của các đồng nghiệp, nhà nghiên cứu đầu ngành trong lĩnh vực IC design, phó giáo sư, Tiến sĩ Trần Xuân Tú, Trưởng phòng TN các hệ thống tích hợp thông minh (SIS Lab) đã đóng góp ý kiến để bài nghiên cứu đạt được thành công nhất định.

Expected results:

Status: accepted and implementing (simulating the Noxim with all cases and configurable scenarios.

I1: (project 1): (priority: normal)

title: Đồng hồ thời gian thực lắp trên xe máy.

Project_name: P1_time_arduino_7Seg4Digi

Introduction:

Sau khi hoàn thiện đề tài này có thể:

- biết điều khiển Led 7 đoạn 4 digit bằng arduino.
- biết quy trình làm mạch in thực tế
- hiểu nguyên lý hoạt động của một số mạch và cách đọc datasheet.
- make money :)
- for fun.

expected results:

- mạch hiển thị thời gian demo trên thực tế điều khiển bằng arduino.
- mạch thực tế gắn trên xe máy để hiển thị thời gian thực qua Led7Seg4Digit.
- báo cáo tổng kết công việc và kinh nghiệm thực tiễn rút ra.

status: developing

#How to make it? Step-by-step.

!#

Pha 1: Thực hiện mạch test thực tế trước.

Step 1: read and do it myself follow this link

<http://www.instructables.com/id/TimeDuino-Arduino-based-clock-using-7-segment-dis/>

username: DIY_vnd

password: normal-number.

Step 2: prepare the devices such as arduino uno, capacitors...

We actually need:

- arduino uno (1)
- ...

Step 3:

next

Pha 2: Triển khai thực tế gắn trên xe máy!

Bước 1: Hoàn thiện mạch cứng, gọn nhẹ, đẹp, bền.

Bước 2: Ra gặp anh thợ tìm cách gắn cho đẹp.

Bước 3: Quảng bá sản phẩm và thương mại hóa.

I2: (project 2): Priority: normal.

title: GPS for security.

Introduction:

expected results:

status: developing

I3: (project 3): Priority: High.

title: Software driver for communicating with ARM platform on Linux Operating system.

Introduction:

expected results:

status: developing

I4: (project 4): Priority: normal.

title: Xây dựng khung chương trình đào tạo ngành ES cho Khoa CNTT&TT; ICTU.

Introduction:

expected results:

status: developing

I5: (project 5): Priority: high.

title: Hardware and software interface on linux.

Introduction:

expected results:

status: developing

I5: (project 5): Priority: high. (SAMPLE)

title: Đồng hồ thời gian thực lắp trên xe máy.

Introduction:

expected results:

status: developing

#Các dự án ý tưởng đang chuẩn bị làm hoặc trong giai đoạn đề xuất chưa duyệt.

#Some ideas and per-proposal projects.

#1: Phương pháp nghiên cứu khoa học.????

#2: Cách đọc tài liệu/báo khoa học?

#3: Các chuẩn giao tiếp phần cứng?

#4: Các chuẩn truyền thông không dây?

#5: Cách viết một bài báo khoa học?

#6: Vấn đề tải cấu hình trên FPGA.

#7: Lập trình vi điều khiển trên hệ điều hành Ubuntu (linux and tools for MCU developed).

#8:

Điện tử tương tự - số cơ bản (the basic of Electronics analog and digital)

#phân biệt “electrical” và “electronic”? The difference between electrical and electronic?

!#

You can read the following link to realize what are the main different points between them

<http://www.brightknowledge.org/knowledge-bank/engineering/careers-in-engineering/electrical-and-electronic-engineering-whats-the-difference>

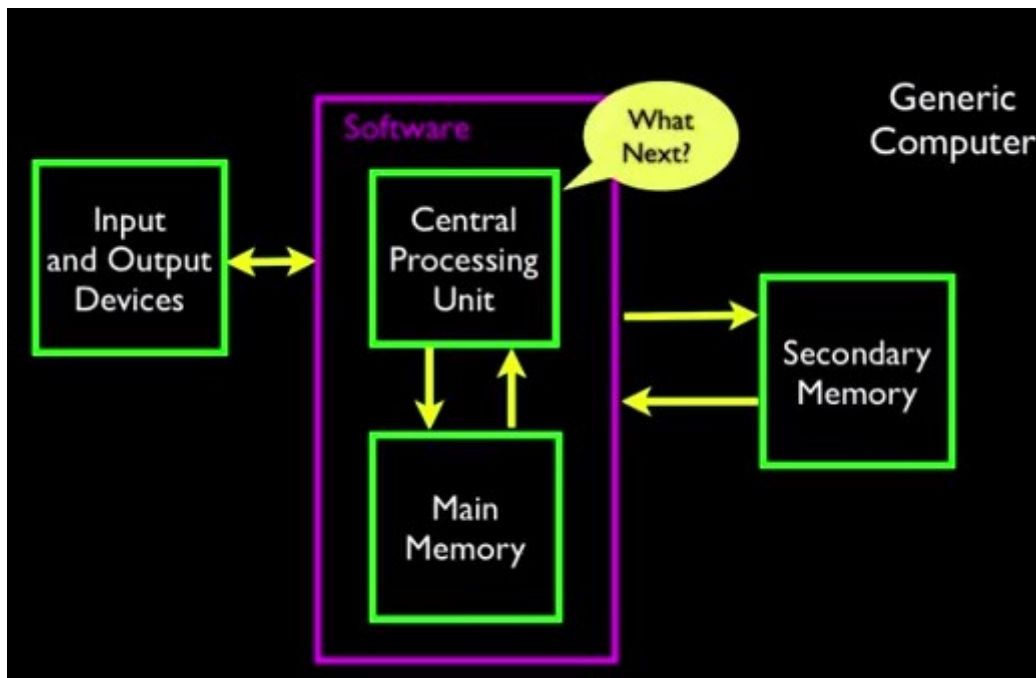
>>>>

#Kiến trúc máy tính (Computer architecture).

Hệ thống kiến thức về phần này tác giả chủ yếu dựa vào những vấn đề khúc mắc trong thực tế nghiên cứu có liên quan cũng như dựa vào tài liệu **Computer Organization and Design: The hardware/ software interface** của nhóm tác giả **David A. Patterson** và **John L. Hennessy** tái bản lần thứ 5 vào năm 2014.

#: Tổng quan về phần cứng.

!#: Tổng quát về phần cứng (Hardware overview) có thể được thấy qua hình vẽ sau: (sưu tầm từ bài giảng về ngôn ngữ lập trình python qua khóa học online!)



Input and Output Devices: Như chuột, bàn phím là thiết bị input đưa dữ liệu vào máy tính, còn như màn hình thì gọi là thiết bị output để hiển thị kết quả và đầu ra của chương trình sau khi thực thi!

Main memory: Bộ nhớ chính ???

Central Processing Unit: Đơn vị xử lý trung tâm.???

Secondary Memory: Bộ nhớ phụ ?? bộ nhớ mở rộng. Ví dụ như RAM, SD card, ổ đĩa cứng... là những nơi lưu trữ dữ liệu tạm thời, không vĩnh cửu.

Nên nghe lại nhiều lần để hiểu những thành phần này là ntn? chức năng là gì? ví dụ... theo link sau: <https://www.coursera.org/learn/python/lecture/YpNQw/lecture-1-2-hardware-overview>

kế đến ???

có thể định nghĩa những khái niệm trên theo hình sau:

Definitions

- **Central Processing Unit:** Runs the Program - The CPU is always wondering "what to do next"? Not the brains exactly - very dumb but very very fast
- **Input Devices:** Keyboard, Mouse, Touch Screen
- **Output Devices:** Screen, Speakers, Printer, DVD Burner
- **Main Memory:** Fast small temporary storage - lost on reboot - aka RAM
- **Secondary Memory:** Slower large permanent storage - lasts until deleted - disk drive / memory stick



#: Các công nghệ dùng để xây dựng bộ nhớ máy tính và các bộ vi xử lý?

!#: Có thể dựa theo hình vẽ dưới đây để nhớ được các mốc ra đời và các mốc hình thành công nghệ thiết kế vi xử lý và bộ nhớ máy tính:

Year	Technology used in computers	Relative performance/unit cost
1951	Vacuum tube	1
1965	Transistor	35
1975	Integrated circuit	900
1995	Very large-scale integrated circuit	2,400,000
2013	Ultra large-scale integrated circuit	250,000,000,000

Transistor là một chuyển mạch bật/tắt được điều khiển bởi đại một tín hiệu điện.

Integrated circuit (mạch tích hợp) là kết hợp hàng chục đến hàng trăm transistor trên cùng một chip đơn.

VLSI (Very Large – Scale Integrated Circuit): Là thiết bị sử dụng tích hợp hàng trăm nghìn đến hàng trăm triệu transistor.

#RISC vs. CISC?

!#

You can read the following links to understand how do they difference others.

<http://cs.stanford.edu/people/eroberts/courses/soco/projects/risc/riscisc/>

others:

Vi điều khiển (Micro-controllers).

#How does a matrix led work?

!#:

#???>>>

#How to start with ARM programming in Linux?

!#

This question maybe met in many cases where someone want to study Programming for Micro-controllers on Linux like me now.

There are some steps for you to answer this question:

Step#1: search on Internet (via google.com.vn) with the keys: ARM programming on Linux, micro-controllers on linux...

I suggest some links:

#for Forum and projects in Micro-controllers and technologies.

#for courses...

Step#2: After you find some suitable links, just try to focus on it to find out more keys and directions to go next.

Step#3: Redo step#1 with other keys when you see some big ideas or hard problems that have to solve to continue.

Step#4: You should try to summary and notes all things that you think is very important to remember and try to review it often as you can.

#How to read a datasheet?

!#

you can read the following link to find out the answer.

<http://electronics.stackexchange.com/questions/16868/whats-this-importance-of-datasheets/16869#16869>

you also can look for other link to read with the keys such as: how to read a datasheet

#How to start micro-controllers on Linux?

#!

Câu hỏi này cũng khá là quen thuộc.

Trước khi đi trả lời nó, suggestion cho một link nói về TOP10 board linux open source để chúng ta tham khảo. Nói chung làm với hệ thống nhúng rồi vi điều khiển trên hệ điều hành mã nguồn mở (Linux) thì theo tôi cần những hệ mã nguồn mở với các kit board như vậy.

Làm các projects trên đó chúng ta sẽ dần hiểu được thế nào là hệ thống nhúng!

<https://www.linux.com/news/embedded-mobile/mobile-linux/732197-top-10-open-source-linux-boards-under-200>

chúc vui!

A comparison of Open source Hardware: Intel Galileo vs. Raspberry Pi.

!# You can read the following link

<http://www.mouser.com/applications/open-source-hardware-galileo-pi/>

#What is a Network Switch vs. a Router?

!#

you can read this link to know more detail

https://www.cisco.com/cisco/web/solutions/small_business/resource_center/articles/connect_employees_and_offices/what_is_a_network_switch/index.html

Summary:

Switches create a network while routers connect to network.

#>>>>

!#

Start with Intel Galileo kit.

#

#Start with Friendly ARM tiny 6400>>??

#>>> community of FriendlyARM.

link for community

<http://www.friendlyarm.net/forum/verify/3019/8a155cd5d6f64e66075c68248473ca5f>

#other link

<http://www.friendlyarm.com/>

username: Van-Nam DINH

email: trailangvocmach@gmail.com

pass: normal.

Should care more about this kit!

#ARDUINO and projects

#Link hoc arduino

<https://www.arduino.cc/>

username: vnd_es

pass: normal

email: namdv2586@gmail.com

commands

!#

cd /opt

sudo chmod a+rw /dev/ttyUSB0

#Projects links and ideas

!#

<http://www.electronicshub.org/arduino-project-ideas/>

#links 18B20 labview and arduino support!

<http://www.hobbytronics.co.uk/ds18b20-arduino>

nice!

#Wifi libraries for arduino

<https://www.arduino.cc/en/Reference/WiFi>

<https://www.arduino.cc/en/Reference/WiFi101>

<https://www.arduino.cc/en/Reference/Ethernet>

<https://www.arduino.cc/en/Hacking/WiFiShieldFirmwareUpgrading>

<https://github.com/arduino/Arduino/tree/master/hardware/arduino/avr/firmwares/wifishield>

<https://github.com/arduino/Arduino/tree/master/hardware/arduino/avr/firmwares>

#>>>>

!#

Mạng trên chip (Network-on-Chip: NoC).

Mạng trên Chip (NoC) là gì? Gồm những thành phần gì? Nguyên lý hoạt động cơ bản như thế nào?

#Cộng đồng phát triển NoC.

#!

Link mạng sau đây sẽ cập nhật các hướng nghiên cứu và các bài báo mới nhất về NoC.

<http://mpsoc.unife.it/~nocsymposium/>

#>>>>

#Một số câu hỏi để Review background của NoC?

!#

- 1) Topo mạng là gì? (nêu khái niệm cơ bản).
- 2) Cấp bậc của bộ định tuyến là gì (router degree)?
- 3) Đường kính mạng là gì (network diameter)?
- 4) Tính quy tắc là gì? (regularity)/ tính đối xứng là gì (symmetry)?
- 5) Tính đa dạng của các đường định tuyến?
- 6)

#Các thành phần của mạng trên chip?

Trong một kiến trúc của mạng trên chip (NoC: Network-on-Chip) gồm các thành phần chính sau:

Các khối chức năng (IP Core).

Các bộ định tuyến (Routers).

Các khối giao tiếp mạng (NI): thực hiện giao tiếp giữa các IP với Router.

What is flow control?

!# Flow control governs the allocation of network buffers and links. It determines when buffers and links are assigned to messages, the granularity at which they are allocated, and how these resources are shared among the many messages using the network.

#Thuật ngữ ánh xạ ứng dụng tiếng anh được biết đến với cụm từ Application mapping có bản chất là gì?

#! Mạng trên chip (NoC) bản chất là một kiểu kiến trúc truyền thông trên chip thay thế cho các *phương thức truyền thông*?? truyền thống như chia sẻ bus (shared bus) hay truyền thông điểm điểm (point to point). Ví dụ như một hệ thống mạng trên chip đã được xây dựng sẵn rồi (một mạng trên chip gồm những thành phần nào có thể xem thêm tại [#NoC's components](#)), sau đó các ứng dụng được đưa vào hệ thống này để đánh giá khả năng truyền thông của mạng. việc ánh xạ ứng dụng tức là với một ứng dụng bất kỳ, đưa vào hệ thống mạng trên chip thì các core trong hệ thống mạng này sẽ tự động thay đổi vị trí phù hợp sao

cho tối ưu nhất về một ràng buộc nào đó mà người thiết kế đặt ra.

Cơ bản về các kiến trúc NoC (NoC architectures)

!#

Phần này có thể tham khảo tốt trong tài liệu (ref1 của SISLab project) Từ trang 35 (48 of 164) đến trang 38 (51 of 164) mục 3.2.1.

Khi đọc xong phần này có thể trả lời một số ý sau:

- kiến trúc chung của một NoC gồm các thành phần chính nào?
- Có các topo nào của NoC phổ biến hiện nay? (mesh, torus...)
- Các tập ô của NoC có ưu/nhược điểm ntn? cái nào được sử dụng phổ biến?
- Truyền thông trong NoC được phân tầng mấy tầng? (7 tầng... giống OSI).

Một số kỹ thuật chuyển mạch và định tuyến trong NoC?

Keys: routing and switching techniques in NoC?

!# ví dụ với một hệ thống NoC gồm các cores là các Ips, rồi chúng được liên kết với các Router (bộ định tuyến) qua giao diện NI, với tập ô là mesh 2D, bây giờ xem nó có những kỹ thuật chuyển mạch và định tuyến nào, đọc tài liệu ref1 trong SISlab projects mục 3.2.2 từ trang 38 để trả lời các vấn đề sau:

- khái niệm routing (định tuyến trong NoC)?
- Khái niệm về chuyển mạch (switching trong NoC)?
- Phân loại định tuyến NoC? (hai loại static and dynamic)??
- Tên một vài thuật toán định tuyến động (minimal adaptive...)?
- Khái niệm *Livelock* và *deadlock* trong NoC?
- có hai kỹ thuật chuyển mạch chính trong NoC? (circuit and packet switching)?
- Nguyên lý truyền thông trong mỗi loại kỹ thuật chuyển mạch và ưu nhược điểm?

Cấu trúc chính của một router sử dụng trong NoC?

Keys: router architectures in NoC?

!#>>> Đọc xong phần này nên giải thích được về kênh ảo (virtual channel)? Khi nào dùng nó, và nó hoạt động như thế nào với một bộ định tuyến chứa kênh ảo? chưa giải quyết được vấn đề này và cần đọc thêm tài liệu (mục 3.2.3 trang 41 hay 54 of 164 của ref1 SISLab project).

Ánh xạ ứng dụng sử dụng thuật toán PSO rời rạc (p159 of 388 NoC book 2015).

!#>>>

PSO được hiểu trong tiếng việt là tối ưu bầy đàn (Particle Swarm Optimization). Được hai nhà khoa học là Kennedy và Eberhart phát triển năm 1995, lấy cảm hứng từ hành vi của đàn chim hay quần thể cá (fish-school) tìm thức ăn. Trong một hệ thống PSO, Nhiều ứng viên có các giải pháp cùng tồn tại và hợp tác một cách đồng thời ngẫu nhiên. Mỗi một giải pháp (solution), gọi là một phần tử (particle), bay?(flies) trong problem ko gian (problem space) qua kinh nghiệm bản thân chúng cũng như kinh nghiệm của các phần tử lân cận (neighborhood particles). Thuật toán (kỹ thuật) PSO được áp dụng thành công giải quyết được nhiều vấn đề trong rất nhiều lĩnh vực. Trong PSO, mỗi giải pháp đơn là một particle trong không gian tìm kiếm, có một giá trị phù hợp (fitness value). Chất lượng của particle được đánh giá bằng độ phù hợp của nó. >>>

XXXXXXXX

#some of problems in NoC. (performance)

!#

- Network performance: should be rapidly and early pre-proved for target applications.
- ...

#abc

!#.

Hệ thống trên chip (System-on-Chip: SoC)

#: Nên bắt đầu học gì khi muốn làm về hướng vi điều khiển?

(How do I start learning microcontroller programming?)

!#: Thực tế câu trả lời cho cái này chung nhất kiểu như là:

1. Learn programming C: Học lập trình C (Nhưng học cái gì ở C vì C cũng nhiều lắm thứ... theo mình là nên bắt đầu với các thứ cơ bản nhất như: xuất nhập trong C, các kiểu dữ liệu, cách viết hàm trong C, rồi mảng và con trỏ trong C...).

2. Start at first with AVR, make some basic project. Bắt đầu với một vài cái dự án cơ bản về AVR. Cái này mình không sure lắm nhưng mình suggest là có thể làm với một vài dự án cơ bản về lập trình cho các dòng vi điều khiển Arduino. dòng VDK nào cũng có điểm chung đó là có tools riêng và bộ thư viện đi kèm, cái này các bạn thấy cái nào hot nhất hiện tại thì cứ việc start với nó thôi. Các từ khóa có thể lên google tra kiểu như là “basic project with arduino” hoặc “basic project with AVR”...

3. You can buy kits from your local market or online.

Cái mục này ý nói bạn nên mua một số các kit hỗ trợ lập trình qua mạng hoặc nơi nào đó gần chỗ bạn sinh sống. Cái này bản chất người viết muốn bạn làm quen với phần cứng (hardware). Đây là một cái theo mình cũng khá quan trọng vì khi chúng ta biết làm một vài project đơn giản với các phần cứng hiện hữu thực tế sẽ hiểu sâu hơn, xem nó có bản chất thế nào và cũng dễ nhớ hơn là điều chắc chắn. (Trực quan sinh động => tư duy trừu tượng mờ). Tùy theo dòng bạn muốn làm nghiên cứu, mà sẽ mua phần cứng đi kèm tốt hơn. hiện tại mình đang nghiên cứu về abc và cũng có support một vài kits hỗ trợ, các bạn có thể tham khảo giá tại một vài nơi, nếu chỗ nào rẻ thì chúng ta mua về học, no problem.!

4. Many websites, forums, blog, are helpful. Cái này công nhận tốt, các bạn có thể tham khảo một vài link sau dành cho dân làm về vi điều khiển như là: a b c, hoặc tốt hơn nữa các bạn nào có thể đọc hiểu E tốt có thể join vào một số các websites, diễn đàn ngoại quốc như a b c thì càng có cơ hội mở mang hơn.

Hey, tiếng anh là quan trọng lắm nhé, vậy nếu bạn nào mà thấy việc học E là cần thiết và muốn nâng cao trình độ ngoại ngữ của mình có thể join vào nhóm do mình lập theo địa chỉ abc sau nhé!

Dân trong ngành còn được có cơ hội trao đổi, thảo luận và chia sẻ thêm kinh nghiệm abc... Thank nhé!

#How to start with IC design field? (What should I study for IC design field?)

!#

A1>> from <https://www.quora.com/What-resources-can-be-used-to-study-practical-IC-design-if-one-cannot-get-internship-outside-school>

For digital ICs, learning Verilog/VHDL is a good training. Using these to program low cost FPGA/CPLD development kits is also wise. HDLs are used to create SoCs and custom digital ICs.

For analog ICs, discrete analog design techniques and circuit analysis is applicable. Design circuits. Build circuits. Debug circuits. (Re)design circuits. Build circuits. Debug circuits. (Re)design circuits. Etc.

The parts missing relate to various bias-independent design and transistor-only design

techniques (no resistors, capacitors, inductors, etc. only transistors). These can at least be simulated in SPIECE even if you don't make IC's themselves.
For analog, the writing of Bob Pease, Jim Willams, etc are essential.

A2>>>>??

SoC trends and applications.
!#

VIM skills

#command for find and changing or replace a word or phrase

!#

:1,\$s/phrase1/phrase2/g

copy and paste

!#

Linux (Ubuntu 15.10)'s commands

Tài liệu chủ yếu đi vào các câu lệnh cơ bản và thường dùng nhất trong quá trình tác giả sử dụng hệ điều hành Ubuntu.

!: Chú ý với bạn đọc rằng tất cả những câu lệnh ở đây đều được tác giả tìm hiểu qua mạng internet, vậy nên trong quá trình học, nếu các bạn dùng google.com để tìm các đáp án thì sẽ có nhiều điều thú vị và nhiều con đường mở ra với các bạn hơn, các giải đáp mình đưa ra cho những vấn đề ở đây đôi khi chỉ là một trong những giải pháp có thể giải quyết được vấn đề chứ chưa hẳn đã là giải pháp hay nhất đâu. Chúc các bạn học tốt! Người thầy lớn nhất hiện tại của mình vẫn là GOOGLE! :)

Một link tham khảo dành cho cách sử dụng câu lệnh cơ bản trong Ubuntu rất hữu ích mà các bạn có thể tham khảo (Tất nhiên với trường hợp không ngại đọc ENGLISH thì tuyệt):

<https://help.ubuntu.com/community/UsingTheTerminal>

Sau đây, tôi xin trình bày cụ thể một vài lệnh tôi hay thường sử dụng trong quá trình làm việc của bản thân!

Câu lệnh check hệ thống/ phiên bản đang sử dụng

!#: một trong những giải pháp có thể thực hiện để giải quyết vấn đề này là gõ phím tắt CTRL+ALT+T để hiển thị cửa sổ gõ lệnh command rồi gõ lệnh sau:

lsb_release -a

kết quả sau khi gõ lệnh sẽ là như sau:

```
namdv@namdv-ThinkPad-T420:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 15.10
Release:        15.10
Codename:       wily
namdv@namdv-ThinkPad-T420:~$
```

ngoài ra có thể dùng lệnh

cat /etc/*release

kết quả cho ra như hình sau:

```
namdv@namdv-ThinkPad-T420:~$ cat /etc/*release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=15.10
DISTRIB_CODENAME=wily
DISTRIB_DESCRIPTION="Ubuntu 15.10"
NAME="Ubuntu"
VERSION="15.10 (Wily Werewolf)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 15.10"
VERSION_ID="15.10"
HOME_URL="http://www.ubuntu.com/"
SUPPORT_URL="http://help.ubuntu.com/"
BUG_REPORT_URL="http://bugs.launchpad.net/ubuntu/"
```

#Linux Basisc

!# https://bash.cyberciti.biz/guide/Main_Page

(Linux Shell Scripting Tutorial (LSST) v2.0).

Lệnh cập nhật hệ thống hay nâng cấp hệ thống.

Câu lệnh này thường xuyên được sử dụng sau một thời gian sử dụng Ubuntu để hệ thống cập nhật và nâng cấp những tính năng mới, công cụ mới cần thiết để hỗ trợ người dùng một cách tốt nhất.

Các bạn vào terminal và gõ lệnh sau:

Lệnh cập nhật hệ thống:

```
sudo apt-get update
```

Sẽ có yêu cầu gõ password đòi quyền admin để cập nhật.

Tương tự như vậy sẽ có lệnh nâng cấp hệ thống, bằng lệnh dưới đây:

```
sudo apt-get upgrade
```

Chú ý hai lệnh trên khoảng vài ngày hoặc một tuần nên chạy một lần, nhất là mỗi lần sau khi bạn cài đặt thêm phần mềm (ví dụ như Vim, gnuplot, code: block,) trên hệ điều hành đang sử dụng!

Check your memory system?

There are some ways, you can use one of the following command:

```
free -m
```

```
cat /proc/meminfo
```

```
vmstat -s
```

Shutdown your computer?

```
sudo shutdown -P 1:00 = shutdown your computer at 1:00 am
```

```
sudo shutdown -c = cancel your shutdown commands
```

how to merge some of the pdf files?

```
pdftk file1.pdf file2.pdf cat output mergedfile.pdf
```

this command was referent form

<http://askubuntu.com/questions/2799/how-to-merge-several-pdf-files>

how to view a pdf file via command line?

!#

```
evince file_name.pdf
```

#Remove file

!#

```
rm filename
```

#Remove folder

```
rm -r /path/foldername
```

```
sudo rm -r -f /path/
```

```
-r: subfolder
```

-f: force

tim den thu muc da cai dat voi wine

~/wine/drive_c/

the above way achieved from

<http://askubuntu.com/questions/354508/how-or-where-will-i-find-c-program-filesx86-in-ubuntu-13-04>

great!

Copy folder in Ubuntu?

!#

#How to extract a file in linux????>>

!#

Checking your OS?

!# type the following command:

lscpu

result:

```
Architecture:          i686
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                4
On-line CPU(s) list:   0-3
Thread(s) per core:    2
Core(s) per socket:    2
Socket(s):             1
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 42
Model name:            Intel(R) Core(TM) i5-2540M CPU @ 2.60GHz
Stepping:              7
CPU MHz:               826.820
CPU max MHz:           3300,0000
CPU min MHz:           800,0000
BogoMIPS:              5183.40
Virtualization:        VT-x
L1d cache:             32K
L1i cache:             32K
L2 cache:              256K
L3 cache:              3072K
```

#>>>>

!#

#GNUPlot

#For nice reference

<http://gnuplot.sourceforge.net/>

#

LaTeX skills

Latex links for sample templates

!#

<http://www.latextemplates.com/>

or

<https://www.sharelatex.com/templates/cv-or-resume>

email: namdv2586@gmail.com

password: normal@123

nice!

#add new fonts to your linux system

!#

You can read the following link

<https://community.linuxmint.com/tutorial/view/29>

for my system, I can do:

```
sudo cp /home/namdv/Documents/SIS\ Lab/Books/0\ Programming/LaTeX\
document/templates/*.otf /usr/share/fonts/truetype/
```

#how to insert a pic into latex file of IEEE conference?

!# you can read the following link to get the answered

<http://tex.stackexchange.com/questions/187548/how-to-insert-an-image-into-latex-ieee-document-by-using-miktex>

position of picture/figure in latex

!# you can ref the following link

https://www.sharelatex.com/learn/Positioning_images_and_tables

after you read the above link, you can understand some of meaning the parameters such as h, t, b, p...

hope it can help.

Note that the homepage of this link is useful for referencing in studying LaTeX

https://www.sharelatex.com/learn/Main_Page

nice!

#Viết tiếng việt có dấu trong LaTeX?

!# Sử dụng link sau:

<http://tex.stackexchange.com/questions/40482/typesetting-any-language-vietnamese-for-example>

có thể copy đoạn code sau tham khảo:

```
\documentclass{article}
\usepackage[utf8]{inputenc}
\usepackage[vietnam]{babel}
```

```
\begin{document}
```

Hà Nội là thủ đô, đồng thời là thành phố đứng đầu Việt Nam về diện tích tự nhiên và đứng thứ hai về diện tích đô thị sau thành phố Hồ Chí

Minh, nó cũng đứng thứ hai về dân số với 6.913.161 người.

```
\end{document}
```

nice!

chèn chính xác vị trí file ảnh trong latex

How to insert an image exactly place in latex.

Keys: “exactly place of figure in latex”

!#

Tham khảo từ link sau:

<http://tex.stackexchange.com/questions/8625/force-figure-placement-in-text>

với hình vẽ chot

```
\usepackage{float}

...

\begin{figure}[H]
\centering
\includegraphics{slike/visina8}
\caption{Write some caption here}\label{visina8}
\end{figure}
```

hehe, ngon!

<<<

#How to use reference in Latex

!#

can read the following link

https://www.sharelatex.com/learn/Bibliography_management_with_bibtex

in introduction part, can copy and run for instance.

#How to use .bib in latex

!#

Programming (C/C++).

#Link for reference

<http://www.cprogramming.com/>

#Classes exercises (These exercises were reference from chapter 9 p389 (417/562)).

Chapter goals:

- To understand the concept of encapsulation.
- To master the separation of interface and implementation.
- To be able to implement your own classes.
- To understand how constructors and member functions act on objects.
- To discover appropriate classes for solving programming problems.
- To distribute a program over multiple source files.

Notes 1 for this chapter:

- A class describes a set of objects with the same behavior. For example, a Car class describes all passenger vehicles that have a certain capacity and shape. You can drive a car by operating the steering wheel and pedals, without knowing how the engine works. Similarly, you use an object through its member functions. The implementation is hidden.
- Every class has a public interface: a collection of member functions through which the objects of the class can be manipulated.
- Encapsulation is the act of providing a public interface and hiding implementation details.
- Encapsulation enables changes in the implementation without affecting users of a class.

Self check 1:

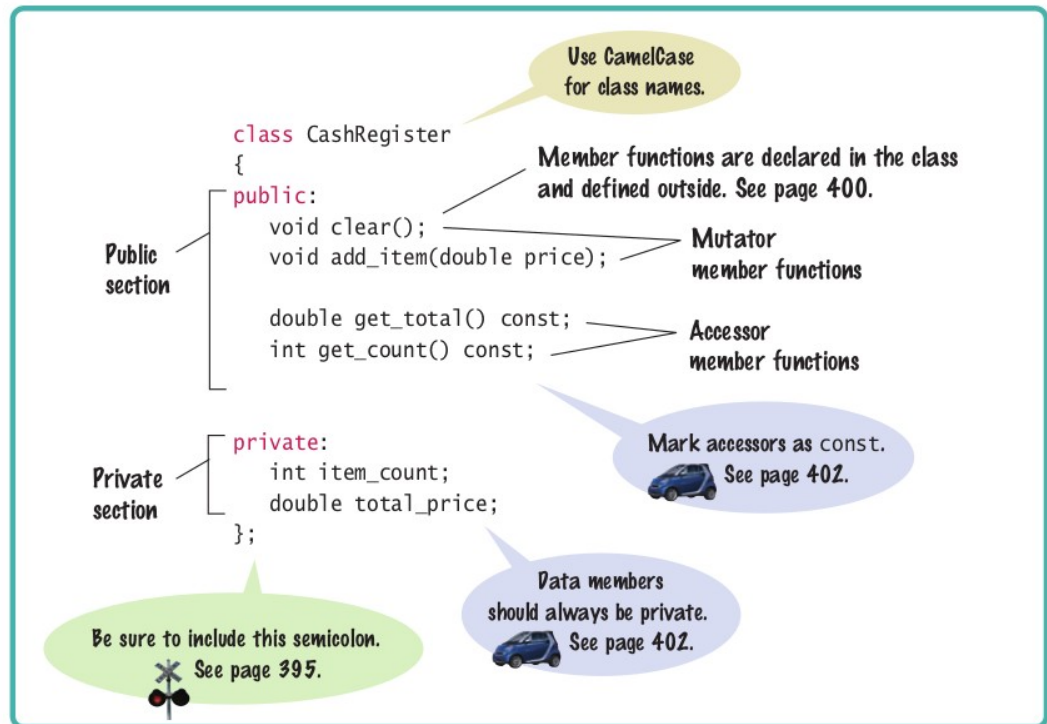
1. In C++, is cin an object or a class? Is string an object or a class?
2. When using a string object, you do not know how it stores its characters. How can you access them?
3. Describe two possible ways in which a string object might store its characters.
4. Suppose the providers of your C++ compiler decide to change the way that a string object stores its characters, and they update the string member functions accordingly. Which parts of your code do you need to change when you get the new compiler?

Notes 2 for this chapter:

- A mutator member function changes the object on which it operates.
- An accessor member function does not change the object on which it operates. Use const with accessors.

See the following figure to understand more in detail.

Syntax 9.1 Class Definition



And

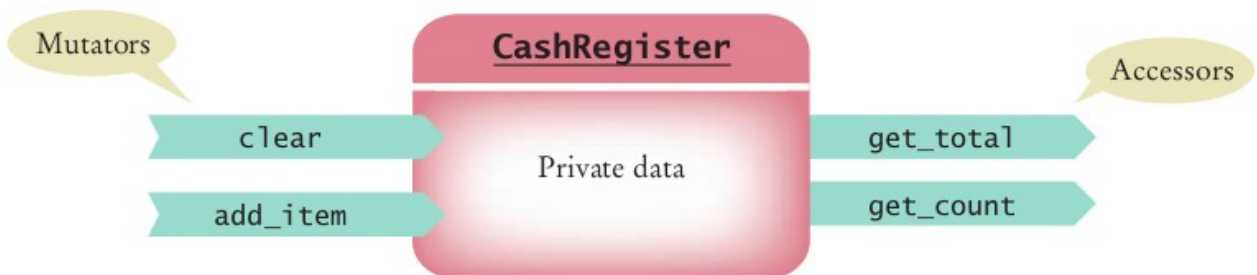


figure: The interface of CashRegister Class.

Check self 2:

5. What does the following code segment print?

```
CashRegister reg;
reg.clear();
reg.add_item(0.95);
reg.add_item(0.95);
cout << reg.get_count() << " " << reg.get_total() << endl;
```

6. What is wrong with the following code segment?

#1: Chương trình đầu tiên “helloworld.cpp”.

!#1: Dùng một trình soạn thảo code như Vim hoặc Text editor trong Ubuntu gõ những dòng code sau và lưu file với tên là *helloworld.cpp*.

```
#include<iostream>
using namespace std;
int main(int argc, char* argv[]){
    cout << "Hello world!\n";
    return 0;
}
```

Sau đó dùng lệnh sau để dịch trong terminal nhé:

```
g++ helloworld.cpp -o test
```

sau khi dùng lệnh trên trong thư mục chứa hiện tại sẽ xuất ra file **test** cho bạn tiếp tục để chạy chương trình trong terminal gõ lệnh sau:

```
./test
```

Như vậy kết quả hiển thị trên terminal sẽ là chữ **Hello world!**

#2: Xuất ra một file txt với dữ liệu lấy từ chương trình.

!#2: Bài toán số 1 giải quyết vấn đề dịch một chương trình trong C++, bây giờ khi ta muốn xuất file ta có thể thực hiện một chương trình đơn giản như sau:

Code mẫu:

```
#include<iostream>
#include<fstream>
using namespace std;
int main(int argc, char* argv[]){
    cout << "Hi, welcome to test output file!\n";
    ofstream myfile;
    myfile.open("example.txt");
    myfile << "writing this to a file.\n";
    for (int i=0; i<10; i++){
        myfile << i << "\t" << i+1 << endl;
    }
    myfile.close();
    return 0;
}
```

sau khi dịch chương trình bằng lệnh:

```
g++ test_out.cpp -o tout
```

kết quả đầu ra sau khi chạy lệnh:

```
./tout
```

như hình dưới đây:

```
writing this to a file.  
0      1  
1      2  
2      3  
3      4  
4      5  
5      6  
6      7  
7      8  
8      9  
9     10
```

chú ý, kết quả hình trên là khi bạn mở file example.txt đã được sinh ra trong thư mục bạn dịch.

(chi tiết các bạn có thể tham khảo thêm theo link:

<http://www.cplusplus.com/doc/tutorial/files/>)

#3. Pointer in C/C++

!#3. Về con trỏ trong lập trình C/C++ ta có thể tham khảo theo link như dưới đây:

https://www.youtube.com/watch?v=h-HBipu_1P0&index=1&list=PL2_aWCzGMAwLZp6LMUKI3cc7pgGsasm2_

Theo như link trên, ta có thể rút ra được các ý chính như sau:

- Nói đến con trỏ thì nhớ đến bộ nhớ (memory như RAM...).
- Biến con trỏ là một biến lưu địa chỉ của vùng nhớ máy tính. Ví dụ khai báo con trỏ kiểu int có tên là p ta có thể dùng lệnh: `int* p;` và truy xuất giá trị mà con trỏ trỏ đến là `&p`.

????? need to read and practice more!

Programming (SystemC/VHDL).

SYSTEMC

Giải thích về sc_main()?

!# Có thể tham khảo trong tài liệu SystemC2.0.1 Language Reference Manual trang 12 (18/428) Mục 5. sc_main() Function.

Có thể tóm tắt đơn giản như sau:

Phần này sau này cần bổ sung thêm sau khi chạy các chương trình mẫu và hiểu bản chất của vấn đề. Tại thời điểm hiện tại khi viết tác giả chưa thực thi code mẫu và chưa hiểu bản chất vấn đề.

Tài liệu tham khảo vừa nêu trên là tài liệu hữu ích trong việc tra cứu về cấu trúc chương trình trong SystemC và giải thích các chức năng cơ bản, khái niệm cơ bản.

VHDL:

Programming (Shell script).
Abc.

Assembly in linux

Link for reference

http://docs.cs.up.ac.za/programming/asm/derick_tut/

#Some advantages when studying assembly!

!# There are some advantages when we try to study ASM:

- It is useful to understand what happens at the lower levels. As you research assembler you will learn about CPU pipelining, branch prediction, cache alignment, SIMD, instruction reordering, and so on. Knowledge of these will help you write better high-level code.
- It's worthwhile to learn a functional programming language, logic programming, scripting language, math-based language...
-

Thường thức cuộc sống và kỹ năng sống.

#ATP viết tắt của từ gì?

!#: Chương trình bảo tồn rùa châu á là từ ATP viết tắt của cụm từ The Asian Turtle Program.
Có thể tham khảo theo link dưới đây:

<http://www.asianturtleprogram.org/>

#OpenCV?

#!

OpenCV là một cộng đồng? mạng ở đó hỗ trợ (support) rất nhiều mã nguồn mở với các ngôn ngữ lập trình khác nhau như C/C++, python... nhằm giúp cho cộng đồng mạng ngày càng phát triển mạnh. Để tham khảo thêm có thể đọc thông tin giới thiệu về hệ thống này theo link dưới đây:

<http://opencv.org/>

have fun!

#What the difference between windows and linux? vnd2586don'tunderstand2

#!

#What is GUI?

!#

GUI stands for **G**raphical **U**ser **I**nterface.

You can know more in detail if you search with “what is GUI” on google.com.vn. It's really cool, I think!

You may get an interesting concept via the following site.

http://www.webopedia.com/TERM/G/Graphical_User_Interface_GUI.html

have fun!

#>>>>>>

!#

#What is Embedded System.

!#:

ES là một khái niệm khá rộng, được dịch sang tiếng việt là “*hệ thống nhúng*”, và được hiểu về bản chất là cái hệ thống nào đó (ví dụ như máy giặt, điều hòa, tủ lạnh, ti vi, vi chip mạch tích hợp...) mà có thể “*nhúng – embed*” vào bên trong nó để thực hiện một chức năng cụ thể nào đó trong cái hệ thống đấy. Vậy về bản chất ta cần phân loại, quan tâm đến hướng nghiên cứu nào trong ES thì là vấn đề cần được đưa ra để thảo luận.

How to focus on my work?

!#: Để tập trung được vào công việc hiện tại của mình bạn nên

- Gạt bỏ những suy tư cá nhân không liên quan.
- Suy nghĩ nhanh, ép não suy nghĩ và trong tình trạng gần như stress để tránh phân tâm.
- Không truy cập các websites xã hội (facebook, twister, hay twoo...) và cũng không nên để điện thoại ON trong khi đang làm việc.
- Trên đây là tự thân, sau đây là tham khảo:

#What the differences between windows 32bit and windows 64bit.

!# ???

#Các thông số chính của một điện thoại mà ta nên biết.

Referencing about reading the mobile phone parameters

http://www.gsmarena.com/lg_g4-6901.php

read and think more.

#What is GCC?

!#

GNU Compiler Collection

read more from

https://en.wikipedia.org/wiki/GNU_Compiler_Collection

#>>>>

!#

#What is OS? What should we know about an OS. (vnd2586don'tunderstand1).

!#

Read the following link to know more some OS (Windows, MacOS, Linux).

<http://www.gcflernfree.org/computerbasics/2>

#???

#What is IDE?

Vns: IDE là từ viết tắt của Integrated Development Environment.

EN: I will look for more information later???

#What are the differences between org, com, ... name domains?

!# You can ref from the following link for getting more information:

<http://www.howtogeek.com/126670/the-difference-between-.com-.net-.org-and-why-were-about-to-see-many-more-top-level-domains/>

Or you can use google to find a suitable link for reading.

#WHAT IS GITHUB?

Link for ref:

<https://github.com/vnd2586>

My account: vnd2586 pass: normal.

!# For this question, you can read in Vietnamese Language according to the following link:

<http://thachpham.com/tools/git-git-va-github-la-gi-tai-sao-nen-dung.html>

For a beginner, you can read the following link to understand more about Github.

<http://readwrite.com/2013/09/30/understanding-github-a-journey-for-beginners-part-1/>

very nice!

>>>>

#What is API in Embedded Systems?

!#

API = Application Program Interface.

these was referenced from the following link

<http://www.webopedia.com/TERM/A/API.html>

then, you can find more with the key “what is API” on google.com to get more Info...>>>

#What is SPI in Embedded Systems?

!#

Serial Peripheral Interface (SPI) is a synchronous serial data protocol used by **microcontrollers** for communicating with one or more peripheral devices quickly over short distances. It can also be used for communication between two **microcontrollers**.

SPI stands for **S**erial **P**eripheral **I**nterface **B**us.

There are some relative links will be added to understand in detail

https://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus

some more links can help you understand more.

<https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi>

<http://www.eeherald.com/section/design-guide/esmod12.html>

<http://www.byteparadigm.com/applications/introduction-to-i2c-and-spi-protocols/>

you can also know about the difference between CAN and SPI via the following link

<http://www.barrgroup.com/Embedded-Systems/How-To/CAN-vs-SPI>

Iddd

#What are the differences between a website and a blog?

!#

The basic difference between a blog and a website is theoretically the difference in how data is being presented – the format.

Ref from <http://www.shoutmeloud.com/difference-between-blog-vs-website.html>

#What are the differences between .net .org .com...

!#

they all are called domain extension such as .net; .org; .com; .info; .biz; .mil; .travel...

.com: Commercial (for-profit) websites.

.net: Network-related domains.

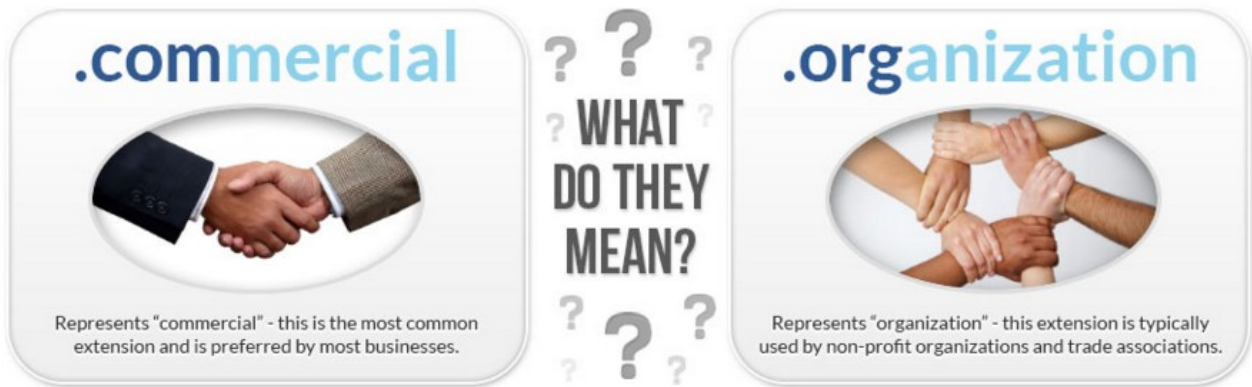
.org: Non-profit Organization.

We can read more information about these from the following link:

<http://www.howtogeek.com/126670/the-difference-between-.com-.net-.org-and-why-were-about-to-see-many-more-top-level-domains/>

this picture was taken from the link <https://itxdesign.com/com-vs-org-which-domain-extension-is-better/>

.com vs. .org, the Differences



.com Domain Extension

.org Domain Extension

#What is PCI?

!#

PCI stands for **P**eripheral **C**omponent **I**nterconnect.

Can read more from this link

<http://www.computerhope.com/jargon/p/pci.htm>

#>>>

!#

#HOWTO:!

#How to start developing for iOS, Android and Windows Phone 7?

!#

Read the following link for referencing.

<http://mashable.com/2010/11/07/how-to-mobile-app-development/#aADv4lus1ZqE>

#>>>

!#

BOOKS and THE BEST REF for my doing research progress.

Embedded System-on-Chip design.

!#

Authors:

Publisher:

Years:

Notes:

Computer Architecture/Organization.

!#

Authors:

Publisher:

Years:

Notes:

Digital System design.

!#

Authors:

Publisher:

Years:

Notes:

Electronic Devices

!#

Authors:

Publisher:

Years:

Notes:

#The most important conversations

Thư thầy Tú gửi ngày 27 January 2016

Dear Nam,

1/ ...

2/ Kiến thức là vô bờ. Mình vẫn luôn phải học, càng học càng thấy thiếu. Nếu em trần trở và thấy thiếu thì đó là điều tốt vì em vẫn còn động lực để học tiếp. Sợ nhất là em thấy đủ/thừa :-D

Tuy vậy, nên có kế hoạch, giữ sức khỏe, cân bằng cuộc sống, **tránh đọc lan man, không dứt điểm được công việc và trở nên hoang mang**. Tập trung xử lý những vấn đề mình nêu để mình trao đổi với em các công việc tiếp theo.

Anh Tài (ngày 13/2/2016 (7/1/2016-âm lịch).

Động lực là do mình tạo ra, có động lực hay hơn là đặt mục tiêu. Mỗi tuần viết một bài tóm tắt hoặc một bài báo cáo có chất lượng chuyên môn tốt, rồi gửi cho Giáo điều dẫn.

Đọc nhiều bài báo sẽ học được những câu phức tạp của người ta, rồi viết nhiều thì thành câu của mình. ban đầu, em có thể chép nguyên si những câu hay, sau đó dần dần sau một khoảng thời gian revise thì sẽ thành của mình. Khi viết nhiều, gửi nhiều, thầy sẽ hiểu những kiến thức và sự phấn đấu của mình. Dù thầy không có thời gian sửa được thì cũng cho mình những ý kiến, đó là động lực nhiều hơn để mình làm việc. Ví dụ, thầy bảo Nam viết tốt đấy, cứ thế phát huy... Vì người ta ngại sửa thôi chứ đọc thì rất thích, vì đọc sẽ được thêm kiến thức, tiếng anh và hiểu thêm được cả sinh viên. Thậm chí sau một loạt bài báo cáo tuần của em, cuối năm tổng hợp lại được bài báo ngon lành. **Mình không cần là người thông minh nhất nhưng hãy là người cố gắng nhất**. Đó là key của sự thành công và sự tín nhiệm. Báo cáo tuần gồm working plan, và cả phần study hoặc research gì đó. Cái phần đó liên quan trực tiếp đến vấn đề mà em đang làm, hoặc em đang thắc mắc. Chắc chắn thầy đọc sẽ biết em thế nào và dần em sẽ tiến bộ theo từng tuần, từng tháng.

Thực tế tình trạng lo lắng quá, dẫn đến đọc nhiều, lan man là có thực. Do bởi nhận thức của mình chưa đến tâm, nên mình vẫn còn lúng túng trong quá trình nghiên cứu, không biết nên làm cái gì trước, cái gì sau?

Về việc đọc/làm gì trước, làm gì sau thì theo kinh nghiệm của anh, một là: Có được các lời khuyên của chuyên gia đầu ngành trong lĩnh vực nghiên cứu của mình (ví dụ như thầy em hoặc những đồng nghiệp giỏi khác). Lời khuyên mình cần là gì? Là cách tìm được các excellent books và excellent papers, và cái gì nên làm trước, cái gì làm sau? Còn làm như thế nào là phải do nỗ lực của mình. Đừng hỏi nhiều quá. Như anh cũng vậy, khi anh đọc được một bài báo hay, anh đọc rất chi tiết, và anh có cảm giác nó như gợi mở cho mình mọi thứ, kha sáng đầu óc mình. Hoặc một quyển sách hay cũng vậy, nó làm cho mình cảm thấy tự tin, kiến thức thu được nhiều hơn, chất lượng hơn. Đọc những quyển liên quan đến phần em nghiên cứu trước, rồi em sẽ có ý tưởng. Căn cứ vào đó để viết báo cáo em ạ.

Cách note quan trọng nhất là viết báo cáo, bởi khi viết là mình phải vào đầu, bút óc để nghĩ và viết thế nào cho logic?... Cái đó sẽ khiến mình nhớ đến chết luôn, một công được mấy việc. Có điều công việc đó vất vả, đòi hỏi phải kiên trì, chăm chỉ nhưng một khi đã vượt qua được rồi thì sẽ thành công.

Một kinh nghiệm nữa anh muốn chia sẻ với N, anh lập một field summary bằng word, rồi cứ đọc được gì hay hoặc nghĩ ra ý tưởng là anh viết luôn vào trong đó. Không quan tâm đến ngữ pháp sai hay đúng, miễn là mình transfer ý tưởng vào cái đã. Người làm khoa học quan trọng nhất là ý tưởng, sau đó đến số liệu và kết quả.

Về vấn đề dịch và giải thích bằng Tiếng Việt không trôi, có time nên đọc sách tiếng Việt, để biết và hiểu cách dùng một vài thuật ngữ. Phải cố gắng chiến thắng bản thân mình. Cố gắng tập trung làm việc hiệu quả cao.

+

#USEFUL WORDS and GENERAL MISTAKES in ENGLISH

#**convince**: /kən'vɪns/ Sự thuyết phục.

N: Sự thuyết phục

V: Thuyết phục

syn: (v) captivate

#**humble**: (adj) khiêm tốn.

#**respect**: (n) sự kính trọng

synonym: [v]: admire.

critical /'krɪt.ɪ.kəl/ : chỉ trích; phê phán.

clinical: lâm sàng

preliminary study: Nghiên cứu sơ bộ.

unveil [v] /ʌn'veɪl/: công bố.

Meaning:

To remove a covering like a curtain from a new structure at a formal ceremony in order to show the opening or finishing of a new building or work of art.

Or

if you unveil something new, you show it or make it known for the first time.

sponsor /'spɒn.sə/: nhà tài trợ; người đảm bảo, người đỡ đầu.

succinct /sək'sɪŋkt/: gọn gàng.

#

EGFLY: English Group For Loving Your-life!

lý do lập nhóm: Muốn giúp mình, giúp người.

help myself, help yourself!

Mục đích:

- Nâng cao sự tự tin giao tiếp.
- Nâng cao kỹ năng thuyết trình.
- Nâng cao khả năng sử dụng ngoại ngữ (English)
- Nâng cao ý thức cộng đồng, xã hội.
- Giúp bản thân và các thành viên có cuộc sống vui vẻ, tốt hơn.

Cách thức hoạt động:

How does it work?

!#:

I will try to construct this group follow some of stages:

Stages 1: Do it by my self: In this stage, I will try by own in studying English (Take and get at least 7.0 IELTS points in 2016).

Stage 2: Building my database (including: referent links, books, skill in listening, reading, writing and speaking.)

Stage 3: Building my own website for manage the resources (links and books...) and members during the course.

Stage 4: Look for someone who have the same idea for developing my goals.

#The link for studying and practicing Irregular Verbs in English

<https://www.ego4u.com/en/cram-up/grammar/irregular-verbs>

LET'S GO!

Stage 1:

Giai đoạn 1: Thăm định độ quyết tâm và lòng tin (qua giao tiếp trong buổi đầu tiên): Giai đoạn này tuy thời gian ngắn nhưng vô cùng quan trọng, nó thể hiện sự quyết tâm theo đuổi mục tiêu của bạn từ buổi đầu tiên đến buổi cuối cùng.

Bạn sẽ làm gì khi?

- + Thấy nản vì nghe không hiểu?
- + Thấy chán vì học không vào?
- + Thấy bất bình vì một sự việc nào đó mà không thể nói ra?
- + ...

Giai đoạn 2: Thông qua các chủ đề (topics) đã được chuẩn bị trước, mọi người luyện ngữ âm, khả năng đọc câu, trình bày ý... giúp đỡ nhau hoàn thiện kỹ năng giao tiếp và trình bày.

Giai đoạn 3: làm song song với giai đoạn 2 là hoạt động ngoại khóa. Thông qua các hoạt động ngoại khóa như tổ chức đến nhà bạn nào đó trong nhóm ở xa xa có cảnh đẹp và thăm đầu đó trong thành phố, nói chuyện, tán gẫu làm tinh thần sáng khoái hơn và hiểu nhau hơn giữa các thành viên trong nhóm.

Giai đoạn 4: Luyện phỏng vấn và xin việc.

Giai đoạn này nhằm đưa ra các bài phỏng vấn thử việc, test các ứng viên xin việc, luyện kỹ

năng giao tiếp ứng xử, kỹ năng tìm việc, viết CV (Curriculum Viate) để tăng cơ hội việc làm trong tương lai.

Giai đoạn 5: Luyện thi chứng chỉ.

Giai đoạn này sẽ luyện thi theo mục đích của từng thí sinh. Đến giai đoạn này sẽ phân làm hai lớp TOEIC hoặc IELTS. có thể liên hệ với giáo viên hoặc người có kinh nghiệm để hợp tác đào tạo.

Ngoài ra các học viên trong nhóm không có điều kiện học theo lớp học hợp tác đào tạo có thể cùng nhóm trưởng học luyện theo hướng dẫn của nhóm trưởng VND!

Topics:

#1: Talk about yourself

- What is your name?
- Are you working or student?
- where do you live?
- your favorite habits?
- what do you do at free time?

#2: Talk about your family?

- give a sentence that you really really want to talk about your family?
- how many people are there in your family?
- what do your father/mother/ brother/sister/ look like?

#3: Talk about your habits?

- what are your familiar activities on week-days?
- what are your familiar activities on the weekend?
- what do you do in free time?
- what do you do when you have a sadness?

#4: Talk about your hometown?

- give a sentence that describes about your hometown?
- What's it's name?
- Where is it?
- What is the specification/specific point when talking about your hometown?
- What things of your hometown proud you?
-

#5: talk about your favorite English song? / Band?

- give a name of song
- can you sing it?
- give some reasons showing why do you like it?

...

#6: Talk about your favorite sport?

- what is the sport name?
- why do you love it?
- do you play it or not?

...

#7: Talk about your favorite food?

- What is it?
- Why do you like it?
- Can you make it? And how?
- do you have any memory with it?

- ...

#8: Talk about your favorite book?

- What is it?
- Why do you like it?
- Who introduces it to you or find it yourself?
- which point is impressed in you in that book?

...

#9: Talk about your neighborhood?

- What is his/her name?
- What does she/he look like?
- What do you study from him/her?

...

#10: Talk about your next vacation?

- do you often have a vacation?
- where do you want to go?
- why do you want to go there?

...

#11:

Outstanding activities.

#1. Where should we go?

- give me an location where you want to suggestion to our group to go to picnic on the next vacation.
- give from 2 to 4 reasons why we should go there?

#2. place A?

#3. Place B?

Skills and opportunities.

#1. How we can become better?

- positive thinking.

...

#2. How to write a CV?

#3. Looking for a job.

#4. How to work more efficient?

#5. How to read a book?

#6: How to read a paper?

#7: How to find the solution?

#8: What should you do when you have a sadness?

#9:

Essays:

#1: Discuss the differences between the way of life in your country now and the way of

life in th past?

A1:

The following is belong to Ahmed Khayam from FutureLearn

I would brainstorm the topic to collect maximum idea about it. Once I have collected my idea, I will outline them as under:

1. Introduction: Life and Change, Changes in Human Society due to different reasons, Past of my country, Present and future
2. Life in old days: No advance technology, without Internet, manual workloads, slow life
3. Present day life: Communication Technology, Internet and its marvels, more fun and amusement, more productivity
4. Comparative analysis: Gains and Losses, Spiritual decay and material gains, Loss of moral values and advantages of Cultural assimilation.
5. Conclusion

My own answer:

In my opinion, there are two main things that can demonstrate the differences between the way of life nowadays and in the past. They are technologies and financing affect.

I would like to focus on two previous factors because of their affections on the human life. First, technologies nowadays is developed very fast, there are a big gap between now and the past. For example, in the past, human have to meet other to make a conversation while we can use Internet or mobile phone to making up a conversation easily. In addition, technologies help people can decrease their time in many tasks such as go by a plane instead of motorbike in the past.

Second, the financing affect has made the society of viewing changing so far from the past to nowadays. The gap between rich man and poor man is bigger and bigger that leads to the losing of balance in society. The more richer in nowadays, the more poorer. But in the past, almost people are the same and all of them are equal.

In conclusion, I intend to 2 factors including changing of technologies and financing development are the most differences between the way of life nowadays and in the past.

aa

finish!

#Note for general English certificates!

#CEFR links.

<http://www.cambridgeenglish.org/exams/cefr/>

<http://www.examenglish.com/CEFR/cefr.php>

#Job Requirements

#useful links for ref.

ENGLISH

#good ref link for studying IELTS

<http://ieltsliz.com/>

<http://www.tolearnenglish.com/>

username: vnd2586

email: namdv2586@gmail.com

pass: bietchetlien2586

<http://www.ecenglish.com/learnenglish/>

#website for checking grammar errors automatically.

<https://app.grammarly.com/>

user: namdv2586

email: namdv2586@gmail.com

password: normal

#>>>>

SOFT SKILLS

#For downloading books

<http://bookzz.org/>

<http://libgen.io/>

<http://book4you.org/>

#for downloading papers via DOI number

<http://sci-hub.io/>

#for downloading from youtube

<http://keepvid.com/>

#ES AND RELATED MAJORS

#Co dien tu

<http://codientu.org/account/>

username: codientu_nam

email: namdv2586@gmail.com

password: [normal@123.](#)

#dien dan PIC viet nam

web: <http://www.picvietnam.com/forum/register.php?do=addmember>

username: PICVN_namdv

email: namdv2586@gmail.com

password: normal

#>>>>

#for asking something

<http://stackoverflow.com/>

user: vnd_es

number of user: **5939238**

email: namdv2586@gmail.com

password: normal

One of the best University that I have known

<http://engineering.missouri.edu/ece/>

beautiful website and fields!

SIS lab

#Website quan ly du an cua phong SIS

<http://10.10.1.61/redmine>

user: namdv

pass: normal

#Website wiki

http://10.10.128.33/wiki/index.php/Main_Page

#Website cua thay Tu

<http://www.uet.vnu.edu.vn/~tutx/>

Online courses

<http://www.memrise.com/home/>

username: namdv

password: normal

<https://www.futurelearn.com/>

email: dvnam@ictu.edu.vn

pass: normal

<https://www.coursera.org/>

email: dvnam@ictu.edu.vn

pass: normal

#ES: Embedded Systems courses online

<http://unex.uci.edu/areas/engineering/embedded/courses.aspx>

HTML and web development

<http://www.w3schools.com/>

<http://thachpham.com/category/web-development/html-css>

#For install a software such as Arduino

sudo apt-get install arduino

For reference GNUPlot skills

<http://gnuplot.sourceforge.net/>

Websites for Embedded systems.

#STM .st

#!

https://my2.st.com/content/my_st_com/en/products/ecosystems/stm32-open-development-environment.html

username or email: dvnam@ictu.edu.vn

password: normal

#ARM community.

<https://community.arm.com/get-started>

username: vnd

email: namdv2586@gmail.com.

password: normal@gmail.com

#link mua ban link kien???

<http://www.reichelt.com/?>

[MWSTFREE=0&CCOUNTRY=447&LANGUAGE=EN&ARTICLE=164977&PROVID=2775&psuma=OEMsecrets](http://www.reichelt.com/?MWSTFREE=0&CCOUNTRY=447&LANGUAGE=EN&ARTICLE=164977&PROVID=2775&psuma=OEMsecrets)

#>>>

#link dien dan dien tu viet nam

<http://www.dientuvietnam.net/forums/>

email: namdv2586@gmail.com

username: vnd_ES

pass: normal

#studying LabVIEW.

web: <https://decibel.ni.com/content/docs/DOC-12287>

user: vnd

pass: normal-number

email: dvnam@ictu.edu.vn

#link tham khao job's requirements

<http://www.electronicweekly.com/>

username: Van-Nam DINH

email: namdv2586@gmail.com

pass: normal

#Link hoc arduino

<https://www.arduino.cc/>

username: vnd_es

pass: normal

email: namdv2586@gmail.com

<https://www-ssl.intel.com/>

username: vnd_es

pass: vnd@es2586

http://www.eeweb.com/register/finish_profile/vannam_dinh

email: namdv2586@gmail.com

pass: normal

<https://github.com/join/plan>

username: vnd2586

pass: normal

<http://www.diendandientu.com/diendandientu/mvnforum/index>

username: learn_electronics

pass: normal-number

#link về điện tử, instructable....

<http://www.instructables.com/>

Username: DIY_vnd

email: dvnam@ictu.edu.vn

password: normal.

#Start with programming for microcontrollers from here

<http://www.tldp.org/HOWTO/Avr-Microcontrollers-in-Linux-Howto/x207.html>

#<http://stackoverflow.com/>

username: vnd_es
pass: normal

#tool for developing AVR on Linux

<http://gputils.sourceforge.net/>

#from website: <http://www.instructables.com/>

username: namdv_es

password: normal.

Email: namdv2586@gmail.com

#web for developing the mobile programming on ubuntu

<https://developer.ubuntu.com/en/apps/>

#web for studying Galileo

!# sparkfun.com

serial: 785799

username:

email: namdv2586@gmail.com

password: normal@123

link: <https://www.sparkfun.com/account/welcome>

from

From their reply

MAR 18, 2016 | 09:38AM MDT

Nick M replied:

Hello-

We have tons of resources for that!

You can start with books: <https://www.sparkfun.com/>

We have kits: <https://www.sparkfun.com/>

We have a huge range of different microcontrollers: <https://www.sparkfun.com/>

And we have an awesome tutorials page as well: <https://learn.sparkfun.com/>

Nick Miranda

SparkFun Electronics

Distributor and Customer Service

303-945-2984 x 607

#Arduino projects

<https://www.hackster.io/users/registration/toolbox>

<https://www.hackster.io/>

username: vnd_arduino

password: normal;

email: namdv2586@gmail.com

#other one for labview arduino links

<https://decibel.ni.com/content/groups/labview-interface-for-arduino>

<<<

#

8) Websites for updating the technology and Job's opportunities.

<http://www.techora.net/2016/02/best-most-profitable-small-business-ideas.html>

#My Dinary

#1: Tổng kết một năm làm việc. (7th March 2016)

Vậy là ở Lab đã hơn một năm ngồi làm việc tại SIS Lab (khoảng tháng 2 năm 2015 mình có buổi gặp và nói chuyện đầu tiên với Thầy!). Đã trải qua quá trình làm việc và học hỏi, mình muốn viết lại một vài ý chính để tổng kết lại những gì đã đạt được, chưa đạt được và cố gắng để tiếp tục phát huy.

Về những gì đã đạt được:

1) Cái đạt được lớn nhất có lẽ là hoàn thành khóa luận tốt nghiệp thạc sỹ với tấm bằng TB Khá!

Dù kết quả cuối cùng không cao nhưng nó là một trải nghiệm vô cùng lớn trong đời mình, là bước ngoặt cuộc đời, 3 năm học mới lấy được bằng Thạc sỹ (từ tháng 9 năm 2012 đến tháng 9 năm 2015).

Với khóa học Hệ thống nhúng (ES) này, mình đã có cơ hội được học và gặp trực tiếp các giáo sư người Pháp, họ dạy mình không hiểu và biết được nhiều kiến thức lắm => điểm thấp trong các kỳ thi, nhưng lại là môi trường để mình buộc phải học Ngoại ngữ (English in advance và beginner of French). Phải nói rằng nếu có cơ hội học một khóa tương tự, chắc chắn mình sẽ cố gắng để kết quả không bê bết như vậy!

2) Cái đạt được thứ hai chính là mình có cơ hội được làm việc với phó giáo sư, Tiến Sĩ Trần Xuân Tú, người mình cảm thấy phục “sát đất” cả về phong cách làm việc lẫn cuộc sống gia đình của thầy. Thầy thực sự là một tấm gương sáng đáng để mình noi theo và học tập. SIS lab hiện tại như một gia đình thứ hai của mình với các anh chị em trong phòng rất quan tâm và sẵn sàng chia sẻ kiến thức, kinh nghiệm - những thứ mà mình thực sự thiếu và cần cải thiện thêm rất nhiều.

3) Cái thứ 3 mình đạt được đó là hiểu rõ hơn câu nói do chính mình đúc rút: “everything will be fine as long as you still want to try”. Trải qua những khó khăn, những khi mệt nhọc với việc ngồi học, đọc... Rồi kết quả cuối cùng thu được (dù chưa perfect) nhưng cũng đã đạt được mục đích. Mình lại càng có niềm tin hơn để tiếp tục cố gắng bước tiếp con đường mà mình đã chọn, - con đường mà ở đó còn biết bao nhiêu thử thách, khó khăn mà không phải ai cũng dám bước đi (đã có nhiều người có dự định đi con đường này nhưng đều đã chọn hướng khác hoặc bỏ cuộc!) Mình có niềm tin là mình có thể làm được. Đây chính là điều mình tâm đắc.

4) Cái thứ 4 mình đạt được đó chính là trong cuộc sống cần chú ý hơn, quan sát hơn, biết đặt lòng tốt đúng chỗ, đúng người, đúng việc. Và điều quan trọng nhất đó chính là phải luôn suy nghĩ về gia đình, pama và những người thực sự đáng quan tâm. Bởi, mỗi người đều có hoàn cảnh khác nhau, khác về điều kiện gia đình, văn hóa sinh hoạt, quan điểm, nhận thức... Do vậy không bao giờ được đồng nhất ai cũng có suy nghĩ giống mình. làm gì cũng nên đặt mình vào người khác để suy nghĩ. Và hãy chú ý, sự thông cảm, sự suy nghĩ chín chắn trước mọi hành động hay quyết định là cách để tiếp tục nâng cao kỹ năng giao tiếp, học hỏi để hoàn thiện bản thân hơn.

Về những điểm chưa đạt được.

Nói về những điểm chưa đạt được thì quá nhiều nhưng nói chung lại sẽ cần hoàn thiện một số kỹ năng cũng như kinh nghiệm sống. Cố gắng trả lời những câu hỏi sau để khắc phục những nhược điểm:

- Làm sao hỏi đúng thầy? đúng câu hỏi?
- Quan tâm cái gì là nên?
- Việc đang làm hiện tại có ý nghĩa gì không? có trong ưu tiên công việc không?
- ...

Về những điều cần cố gắng thực hiện trong năm 2016.

Thực tế trong bản lập lịch (scheduling) mình đã nhiều lần đề xuất cái vấn đề này rồi, nói chung có hai kế hoạch lớn cần đạt trong năm 2016 đó chính là thi chứng chỉ IELTS (tối thiểu 7.0) và hoàn thành bảo vệ đề cương NCS do thầy Trần Xuân Tú hướng dẫn đúng hạn.

Để thực hiện được hai mục đích này thì cần đòi hỏi cá nhân mình cần nỗ lực làm sao để tập trung cao độ, đạt hiệu quả cao trong công việc, còn về timetable thì theo cá nhân mình nhận định là khá ổn với 3 khung giờ chính trong ngày từ thứ 2 đến thứ 6 (thứ 7 và chủ nhật sẽ review vào buổi tối và dành riêng time quan tâm gia đình). 3 khung giờ chính là 5-8 (English); 9-17 (SIS lab) và 20-22 (Interested things).

#2: Nhật ký về sự trải nghiệm:

!# Trải nghiệm tuyệt vời!



Nam Đình Văn ▶ Public

1s

Ngày hôm nay

- Một niềm vui mới (dù trải nghiệm này đã từng thử nhưng cũng thấy khác)
 - Một sự hưng phấn không hề nhẹ (trong công việc).
 - Một sự giác ngộ không dễ quên (chuyên ngành!)
- note! Nhật ký số 2!

- Một niềm vui mới: dù trải nghiệm đã qua nhưng cũng thấy run người sung sướng khi biết tin Zin (nhà Cường) trào đời, chú chúc con mau ăn chóng lớn và sẽ sớm trưởng thành, ngoan ngoãn để pama và gia đình con luôn vui vẻ hạnh phúc!
- Một sự hưng phấn không hề nhẹ: Thật căng thẳng khi lúc này đây, các dự án (projects) cả tự chế lẫn ép buộc là khá căng và nhiều khiến cho tâm lý không ổn định lắm (lo ngại). Nhưng những điều này rồi sẽ qua thôi, mình biết và tin vào điều đó vì đã nhiều lần mình gặp phải và với độ lý mình luôn vượt qua. Mục tiêu trước mắt là báo cáo tuần tới về NOXIM giải thích kết quả cuối cùng trước thầy và ace trong phòng SIS Lab, sau đó đến mục tiêu báo cáo ngày 8/4 trên bộ môn về seminar chủ đề đã đăng ký. Ngoài ra việc business (làm thêm của mình cũng tiến triển tốt) làm cho mình an tâm hơn về tài chính (dù chưa được đồng nào). Các khoản nợ sẽ dần được xóa bỏ với sự chăm chỉ mà mình dành ra cho nó. Niềm tin vô đối!
- Một sự giác ngộ không dễ quên: đó chính là khi đọc về NOXIM config. mình nhận ra rằng thật thú vị với câu nói thầy vẫn nói, sẽ trưởng thành hơn khi hoàn thành một dự án, và bài báo lần này, không biết kết quả có đăng tốt hay không, nhưng mình tin hơn rằng mình đang trưởng thành trong hướng đi đã chọn. Có niềm tin là có tất cả!

Interesting myself!

#3:

abc??>>>>

ABBREVIATION and explanation.

ACM: Association for Computing Machinery. (?)

ASIC: Application Specific Integrated Circuit.

BSD: Berkeley Software Distribution.

CAN: Control Area Network.

DCE: Data Communication Equipment.

DOR: Deterministic dimension Ordered Routing.

DTE: Data Terminal Equipment.

DTR: Data Terminal Ready.

DVOPD: Dual Video Objective Plane Decoder Application

ES: Embedded System.

GALS (Globally Asynchronous - Locally Synchronous): Hệ thống đồng bộ cục bộ - dị bộ toàn cục (ref from prof.Tu's book).

IDE: Integrated Development Environment.

IEEE: The Institute of Electrical and Electronics Engineers.

IP: Intellectual Property.

ITU: International Telecommunication Union: Liên hiệp viễn thông quốc tế.

LLC: Logical Link Control.

MCU: Micro-Controllers Unit.

MEMS: Micro-Electro-Mechanical Systems.

MPEG: Moving Picture Experts Group.

NI: Network Interface: Giao diện mạng.

NoC: Network-on-Chip: Mạng trên chip.

OBDD: Ordered Binary Decision Diagrams.

OSI: Open System Interconnection: Mô hình tham chiếu liên kết các hệ thống mở.

PIR: Packet Injection Rate

PWM: Pulse Width Modulation.

SPI: Serial Peripheral Interface

TWI: Two-wire Serial Interface. (?)

USART: Universal Synchronous and Asynchronous Serial Receiver and Transmitter.

VCEG: Video Coding Experts Group.

VHDL: Very High Speed Integrated Circuit Hardware Description Language.

VOPD: Video Objective Plane Decoder Application.

Bibliography

#LTE: Long-Term Evolution

This is a standard for wireless communication of high-speed data for mobile phones and data terminals.