电力电子编程入门

王赛

2019.08.31

概述

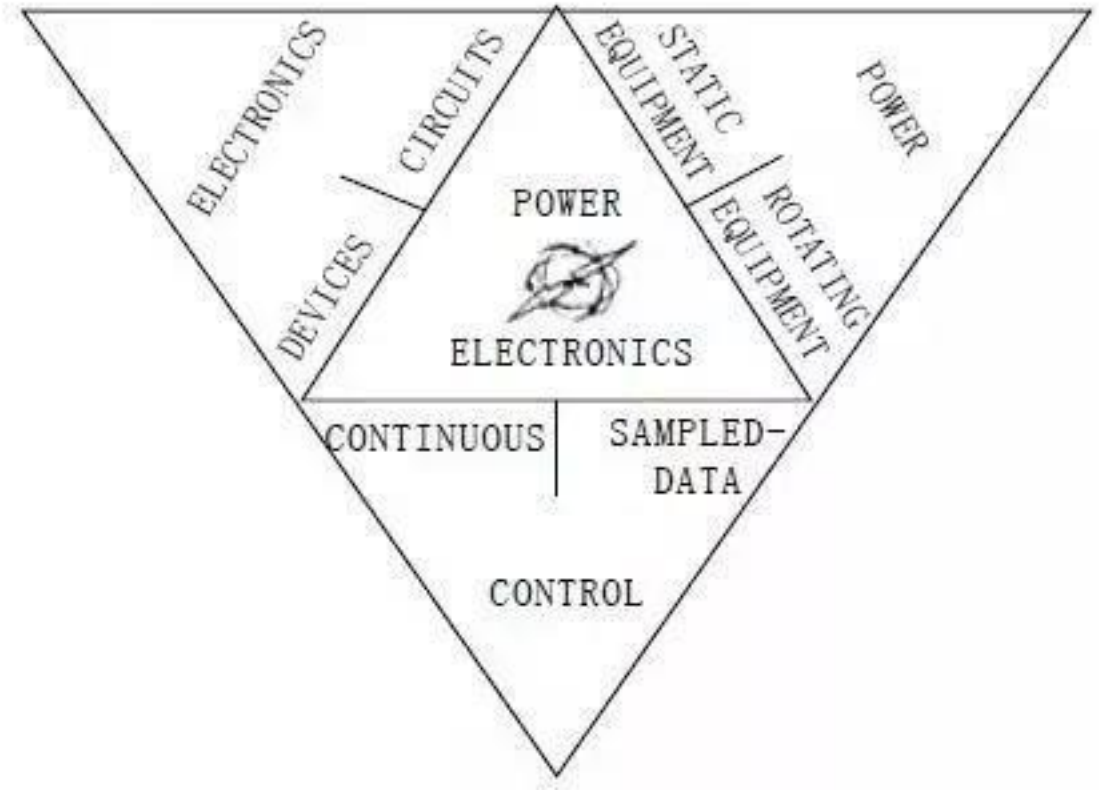
- 培训对象：准研一学生
- 预备知识：基本的C语言和DSP（TI-C2000系列MCU）
- 内容：电力电子编程学习路线图+工作学习方法
- 原则：

（1）**系统和宏观的概述**，介绍学习方法和路径，或者介绍入门参考资料

（2）不会进行特别详细的讲解和演示，主要靠自学

第一部分：电力电子编程学习路线图

电力电子学是交叉学科
——电力学、电子学和控制理论



电力电子学廣泛涵蓋電路學、电子学、控制理論、電磁學、信號處理、電力系統、電機機械、半導體物理學等多科學門，但以美国威廉·尼威爾所提出：电力电子学是由电力学、电子学和控制理论三个学科交叉而成的观点獲全世界普遍認同[3]。

编程在电子电子专业中的位置

开发流程

① 产品定位/需求分析

② 调研论证/可行性分析

③ 硬件设计（主电路和控制电路）

④ 结构设计（器件布局和散热设计）

⑤ 软件设计（控制软件和监控软件）

⑥ 集成装配

⑦ 试验测试

知识结构

① 电路拓扑的原理与设计

② 仿真分析（Matlab/Simulink）

③ 元器件选型

④ 主电路结构设计

⑤ 控制电路设计

⑥ 控制软件设计（DSP）

⑦ 技术文档/学术论文撰写技巧

电力电子编程与传统IT编程的区别

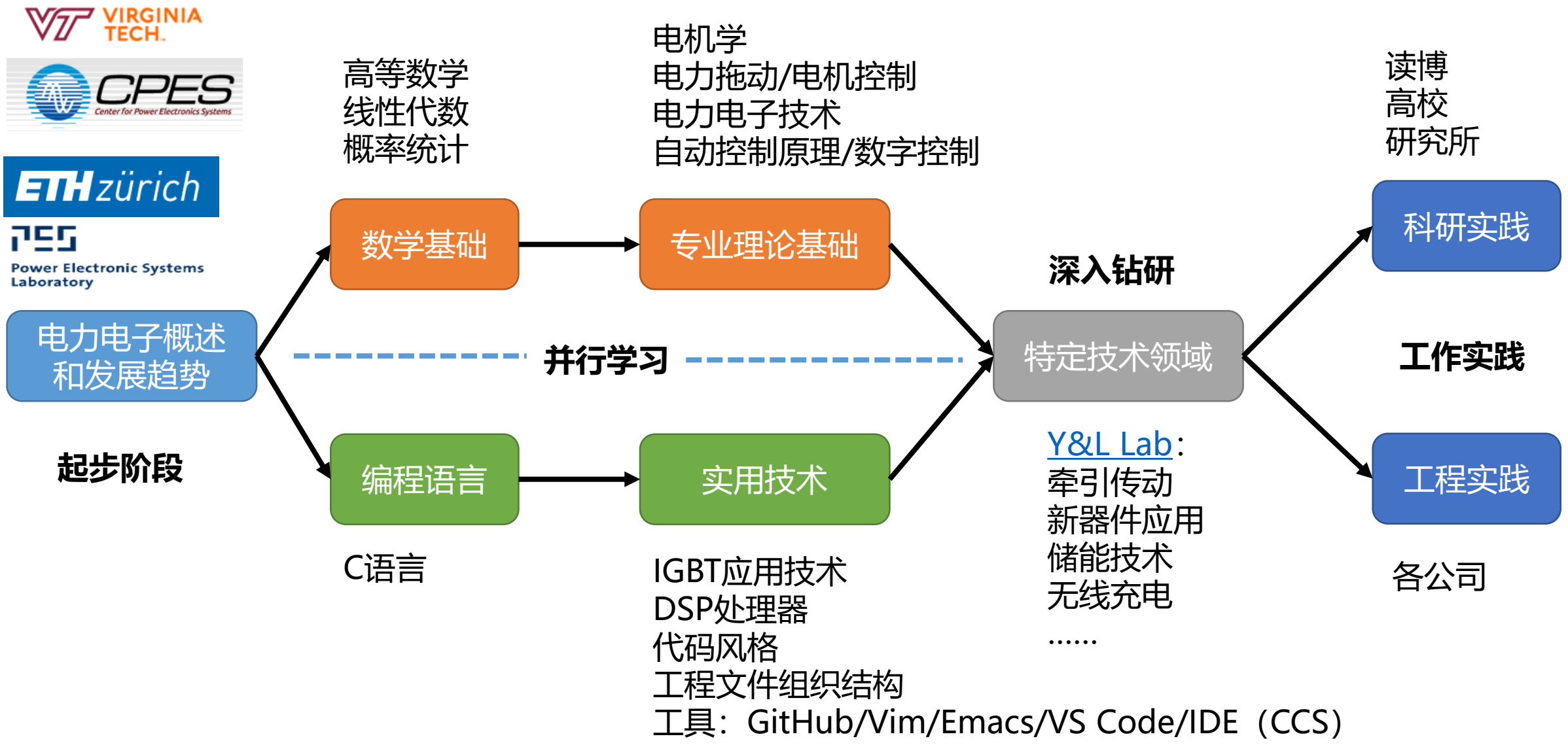
	电力电子编程	计算机专业
数学基础	高数、线代、概率	高数、离散数学、线代、概率
专业基础理论	电机学、电力电子技术、高电压、电力系统分析、自动控制原理	数据结构与算法、计算机组成原理、操作系统、计算机网络
编程语言	C语言	C/C++、Java/C#、Python、JavaScript
实用技术	嵌入式控制器（DSP、ARM）、PI控制器、坐标变换、实时操作系统（RTOS）	各种流行的开发框架和工具

◘ 电力电子编程特点

- 偏底层硬件：DSP控制器、寄存器操作、抽象程度低
- 编程技术在电子电子学中的应用：计算机科学与电力电子学的交叉部分

★ **延申阅读Why Software Is Eating the World—— Marc Andreessen**

电力电子编程学习路线图

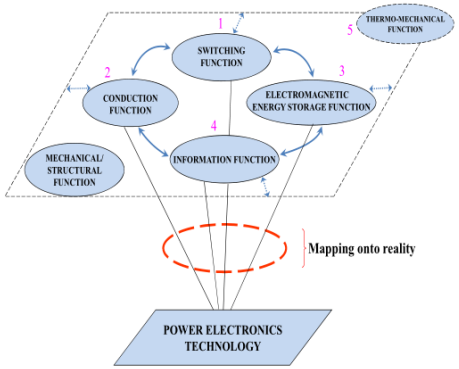


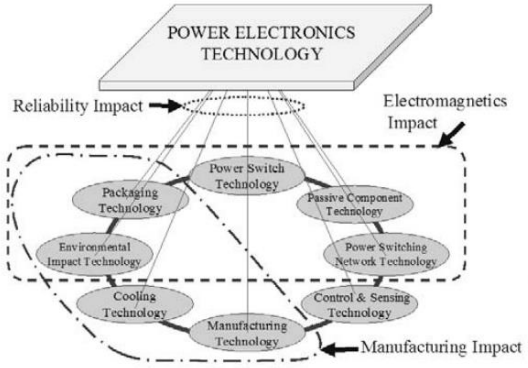
第二部分：特定课程学习建议

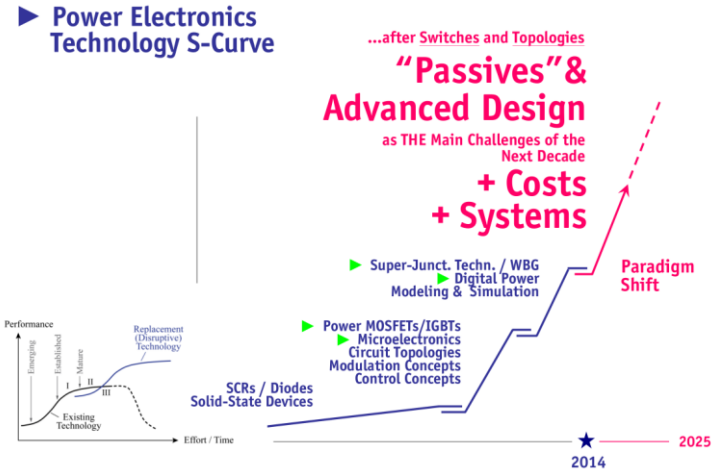
电力电子概述和发展趋势

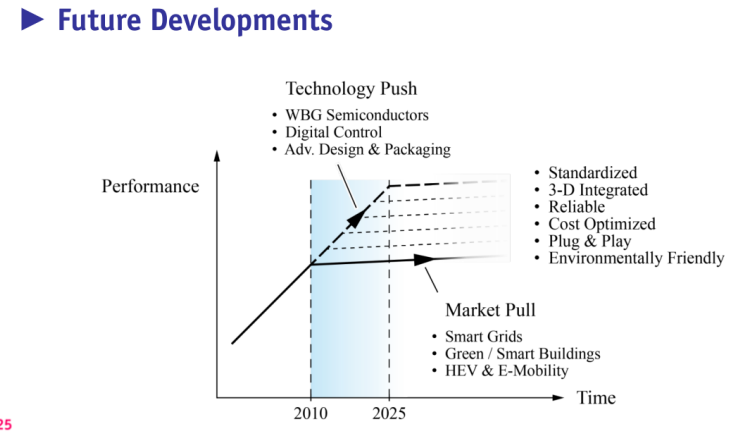
- On a Future for Power Electronics
- 方向：新应用+外部组成技术

- Future Challenges for Research and Teaching in Power Electronics
- 演讲视频Youtube
- 方向：新器件+集成+新应用+多目标优化









专业理论基础

- PI控制器—位置式和增量式，相应的C语言实现
- 坐标变换—矢量控制中最重要的概念，没有之一

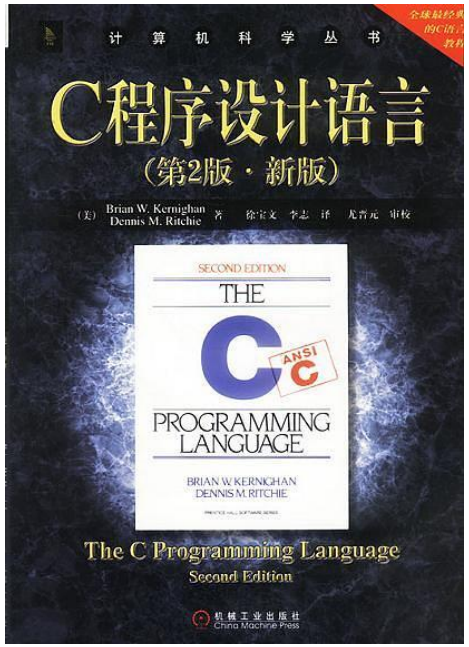
```c
// PI调节器结构体
struct PiController {
  float p_factor;  // 比例系数
  float i_factor;  // 积分系数
  float t_sample;  // 采样周期

  float error_max;
  float increment_max;
  float output_max;

  float reference;
  float feedback;
  float error;
  float error_old;
  float output;

  float integrator;//加了一项
};
```

```c
pi_iq_current.reference = 0.0;
pi_iq_current.feedback = axis_current.q_filter;
PiLoop(&pi_iq_current);
```

```c
// PI调节器函数
void PiLoop(struct PiController *var) {
  float temp, ptemp, itemp, outtemp;
  // 本次误差
  temp = var->reference - var->feedback;
  // 本次误差增量限幅
  if (temp > var->error_max) temp = var->error_max;
  if (temp < -var->error_max) temp = -var->error_max;
  var->error = temp;
  // 本次增量
  ptemp = var->p_factor * (var->error - var->error_old);  //比例项增量，增量式算法
  itemp = var->i_factor * var->t_sample * var->error;  //积分项增量
  // 本次增量限幅
  temp = ptemp + itemp;
  if (temp > var->increment_max) temp = var->increment_max;
  if (temp < -var->increment_max) temp = -var->increment_max;
  // 输出
  outtemp = var->output + temp;
  // 输出限幅
  if (outtemp > var->output_max) outtemp = var->output_max;
  if (outtemp < -var->output_max) outtemp = -var->output_max;
  var->output = outtemp;
  // 保存本次误差值
  var->error_old = var->error;
}
```

编程语言-C语言

- 完整覆盖C99超越了K&R。
- 内容丰满，中文版600页
- 习题质量平均水准比较高提供PPT讲义和在线教师资源
- 探讨现代编译器的实现，揭穿了各种古老的C语言神话和信条
- 前12章基础内容，满足电力电子编程的基本要求

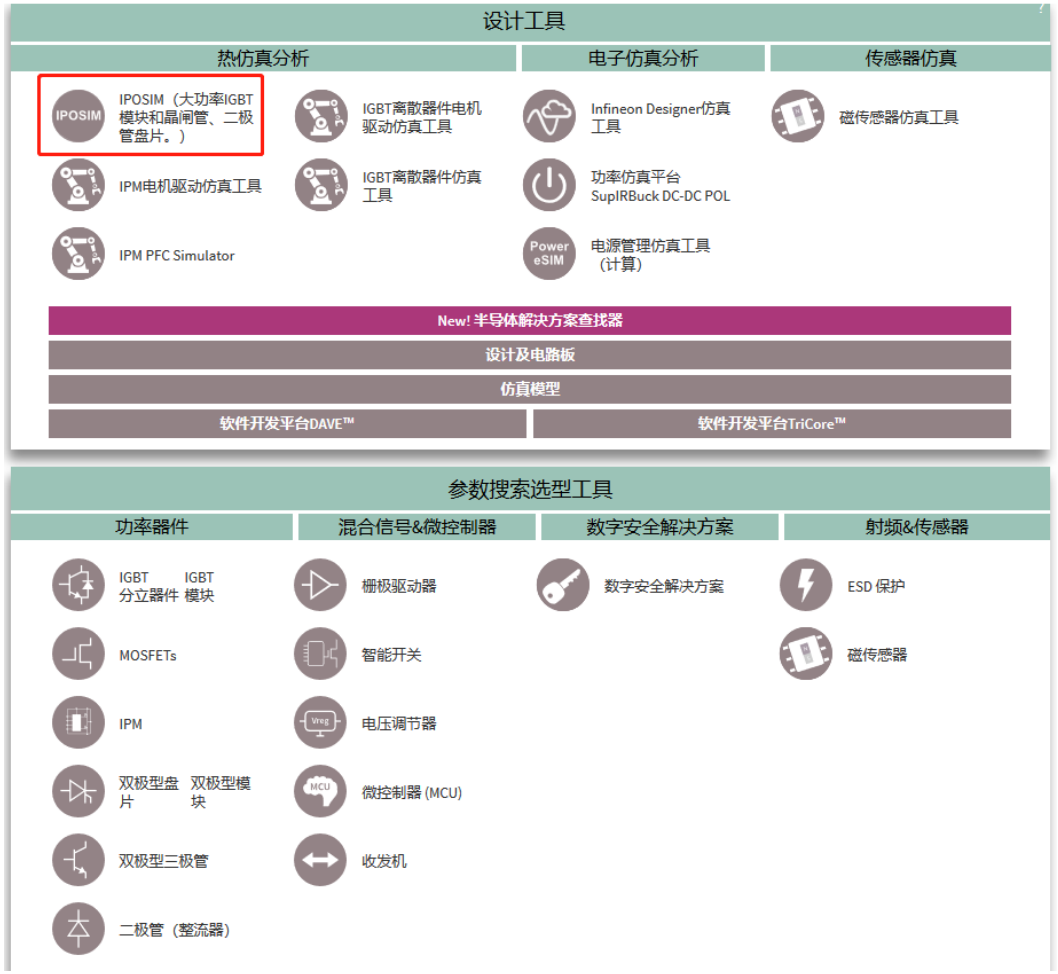


- C语言圣经，简称K&R
- 语言简洁优美，技术作家的标杆
- 适合进阶，或者收藏

- 翻译：以程序员的视角理解计算机系统
- CMU计算机导论，简称CSAPP
- 一本书，树立计算机系统全局观

[CSAPP阅读建议]

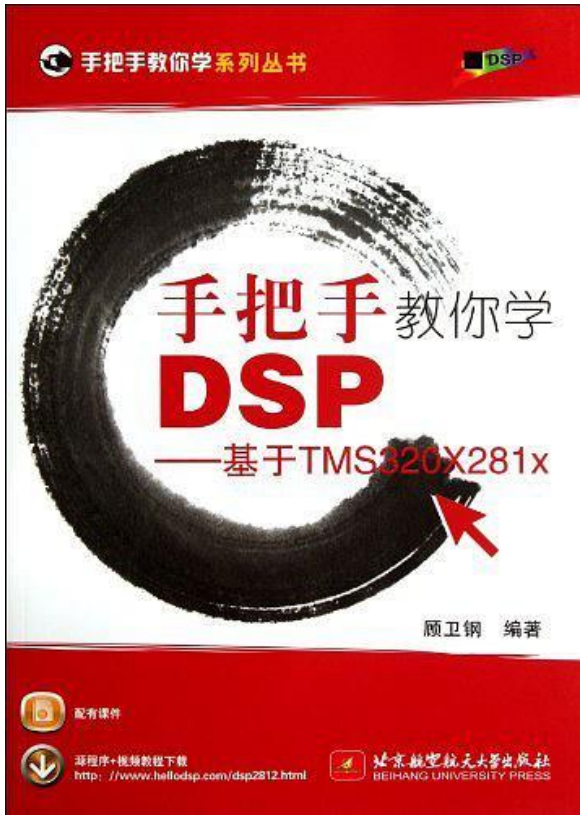
实用技术—IGBT应用技术



- 英飞凌在线工具使用指南

- 作者为英飞凌技术专家
- 英文最新版为第三版
- 中文最新版为第二版

实用技术—DSP处理器



- 入门资料

28335数据表.pdf
CCS5.5的详细操作说明.pdf
IQmath_Quickstart.pdf
TMS320C28x Assembly Language Tools User's Guide.pdf
TMS320C28x Floating Point Unit and Instruction Set Reference Guide.pdf
TMS320C28x Optimizing C C++ Compiler User's Guide(spru514c).pdf
tms320f28335.pdf
TMS320x28x, 28xxx Serial Peripheral Interface (SPI) Reference Guide (Rev. D).pdf
TMS320x28xx 28xxx DSP Peripheral Reference Guide(spru566i).pdf
TMS320x28xx, 28xxx Enhanced Capture (ECAP) Module Reference Guide (Rev. A).pdf
TMS320x28xx, 28xxx Enhanced Controller Area Network (eCAN) Reference Guide (Rev. E).pdf
TMS320x28xx, 28xxx Enhanced Pulse Width Modulator (ePWM) Module (Rev. C).pdf
TMS320x28xx, 28xxx Enhanced Quadrature Encoder Pulse (eQEP) Module RG (Rev. A).pdf
TMS320x28xx, 28xxx Inter-Integrated Circuit (I2C) Reference Guide (Rev. A).pdf
TMS320x28xx, 28xxx Serial Communications Interface (SCI) Reference Guide (Rev. B).pdf
TMS320x280x High-Resolution Pulse Width Modulator (HRPWM) Reference Guide (Rev. A).pdf
TMS320x280x to TMS320x2833x or 2823x Migration Overview.pdf
TMS320x2833x Analog-to-Digital Converter (ADC) Reference Guide.pdf
TMS320x2833x Boot ROM.pdf
TMS320x2833x Direct Memory Access (DMA) Reference Guide.pdf
TMS320x2833x External Interface (XINTF) Reference Guide.pdf
TMS320x2833x Multichannel Buffered Serial Port (McBSP) User's Guide.pdf
TMS320x2833x System Control and Interrupts Reference Guide.pdf

- 数据手册（总体介绍）
- 技术参考手册（各个外设及寄存器介绍）
- TI官方技术支持论坛

实用技术—代码风格

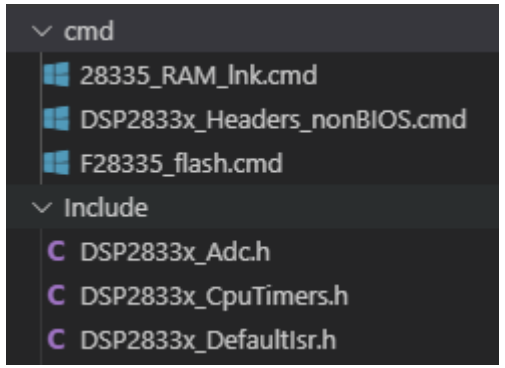
“程序必须为阅读它的人而编写，只是顺便用于机器执行。”

————Harold Abelson 和 Gerald Jay Sussman，《SICP》作者

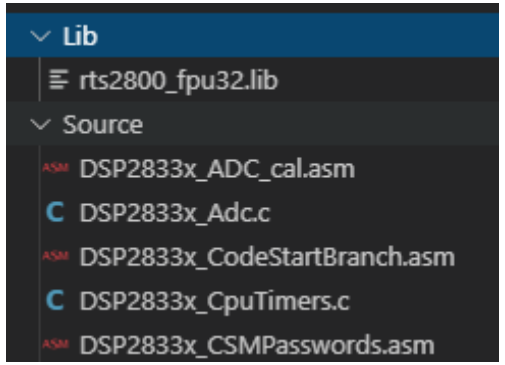
★ 嵌入式C语言风格指南

- **进阶：MISRA-C:2004**
- **其他：华为C语言编程规范**

实用技术—工程文件组织结构



- CMD：分配存储空间，指定寄存器地址
- 头文件：外部接口声明，如函数声明、宏定义、类型定义



- 库文件：可执行代码，如TI官方的浮点运算库
- 源文件：代码

□ 例子：DSP28335模板工程

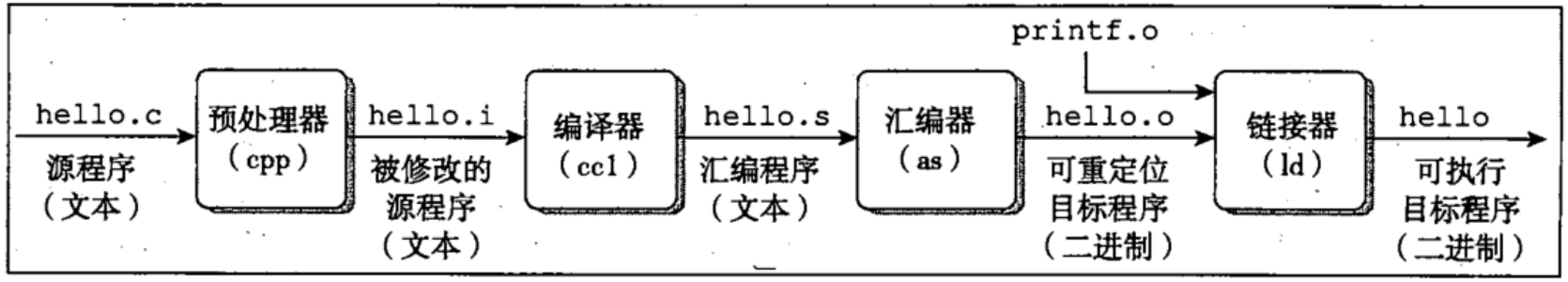


图 1-3 编译系统

实用技术—GitHub

关于版本控制

什么是“版本控制”？我为什么要关心它呢？版本控制是一种记录一个或若干文件内容变化，以便将来查阅特定版本修订情况的系统。在本书所展示的例子中，我们对保存着软件源代码的文件作版本控制，但实际上，你可以对任何类型的文件进行版本控制。

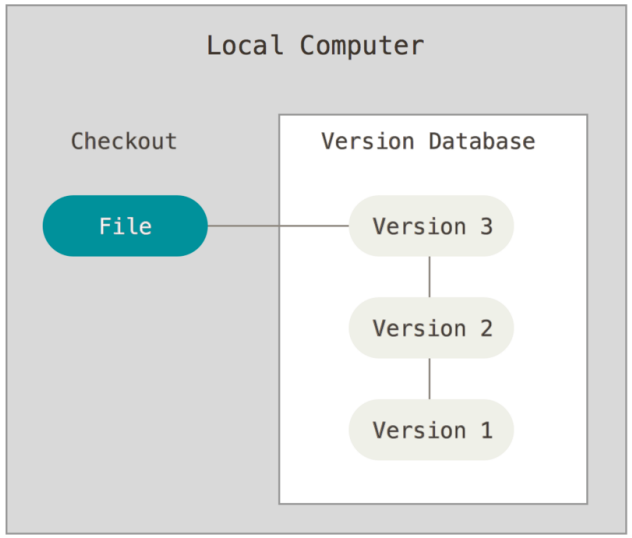


Figure 1. 本地版本控制.

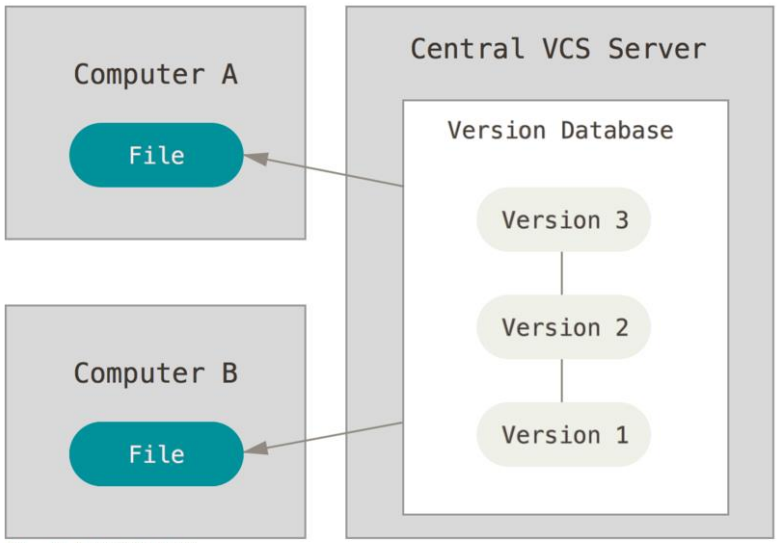


Figure 2. 集中化的版本控制.

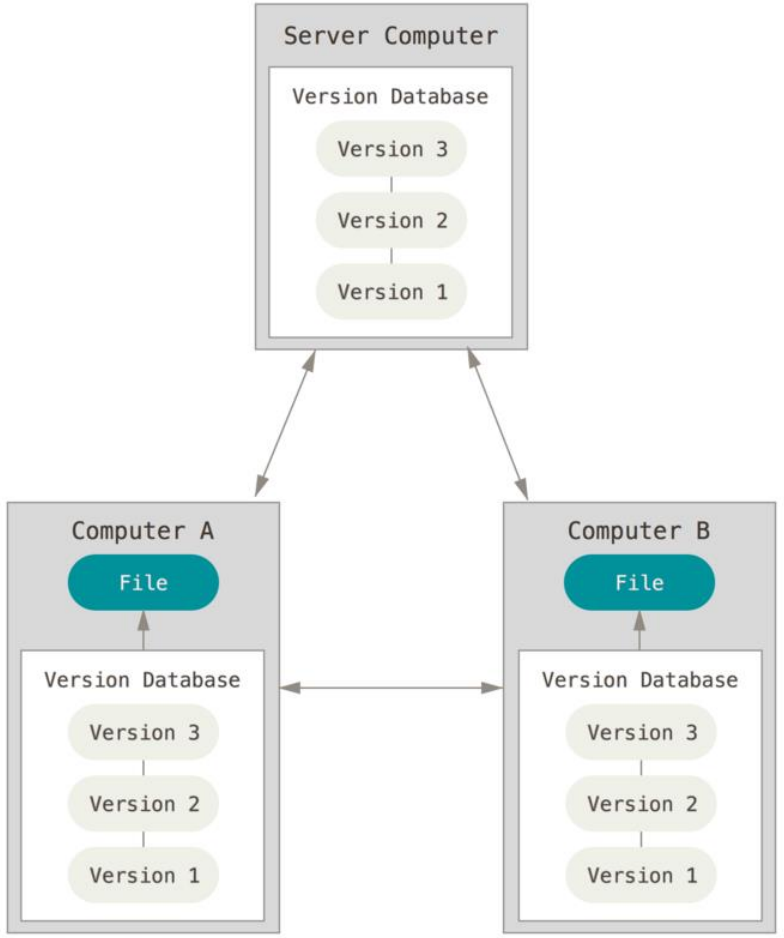
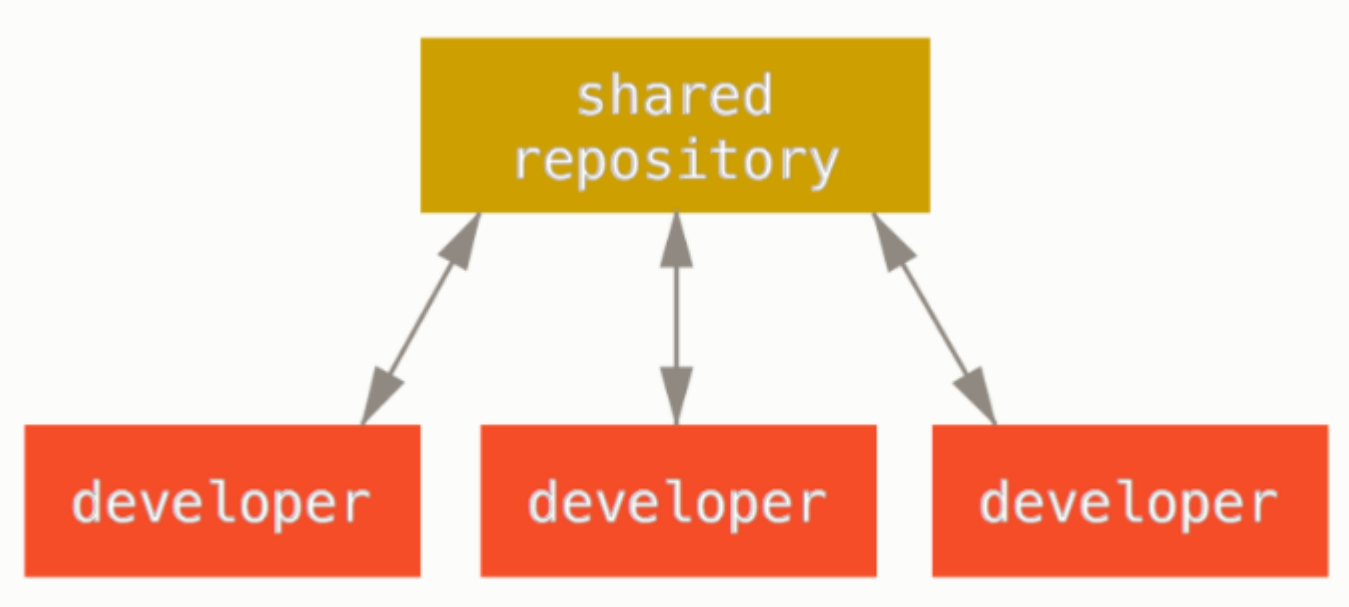


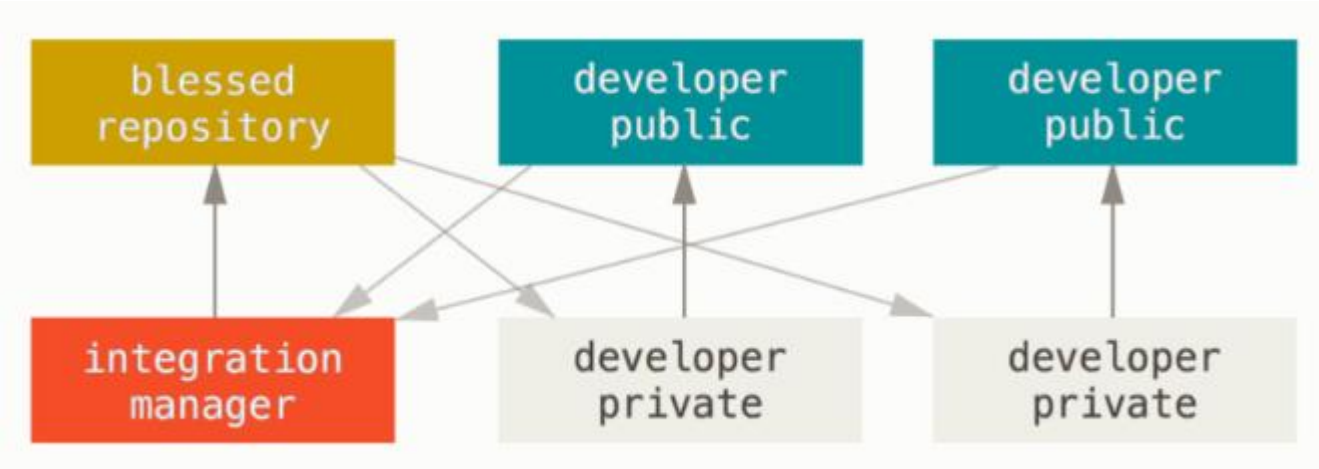
Figure 3. 分布式版本控制.

工作流（work flow）——集中式



- ✓ 如果两个开发者从中心仓库克隆代码下来，同时作了一些修改，那么只有第一个开发者可以顺利地把数据推送回共享服务器。
- ✓ 第二个开发者在推送修改之前，必须先将第一个人的工作合并进来，这样才不会覆盖第一个人的修改

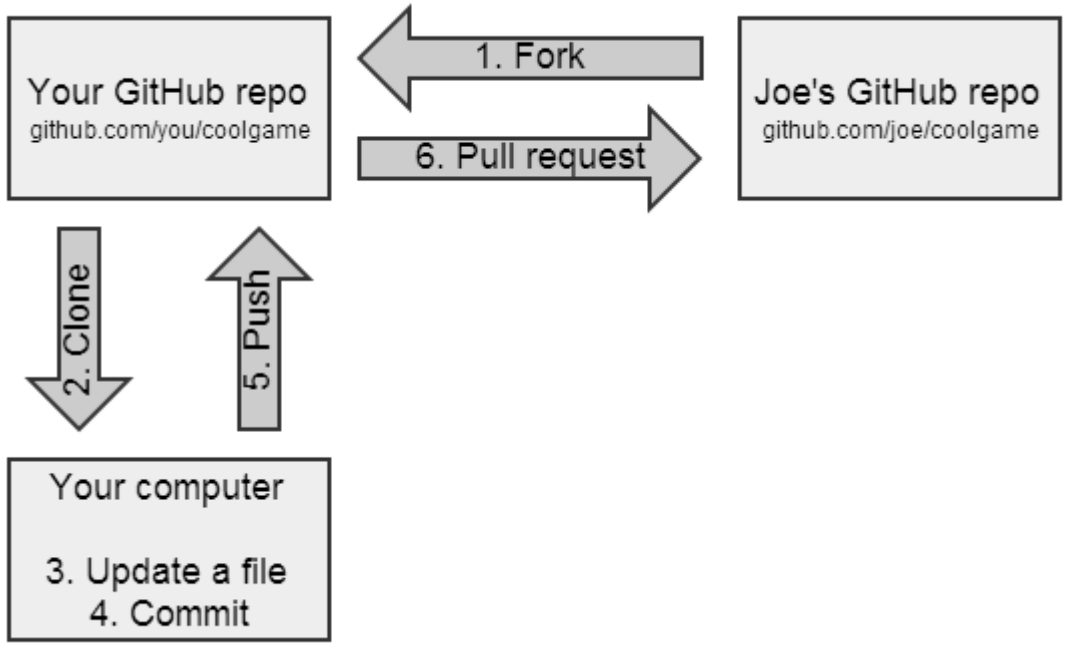
工作流（work flow）——集成管理者



1. 项目维护者推送到主仓库。
2. 贡献者克隆此仓库，做出修改。
3. 贡献者将数据推送到自己的公开仓库。
4. 贡献者给维护者发送邮件，请求拉取自己的更新。
5. 维护者在自己本地的仓库中，将贡献者的仓库加为远程仓库并合并修改。
6. 维护者将合并后的修改推送到主仓库。

最主要的优点之一是你可以持续地工作，而主仓库的维护者可以随时拉取你的修改。贡献者不必等待维护者处理完提交的更新——每一方都可以按照自己节奏工作

集成管理者——Fork并更新一个仓库



1. Fork他的仓库：这是GitHub操作，这个操作会复制Jo
些东西）。复制后的仓库在你自己的GitHub帐号下。

Clone你的仓库：这是Git操作。使用该操作让你发送"
现在这个仓库就会存储在你本地计算机上。

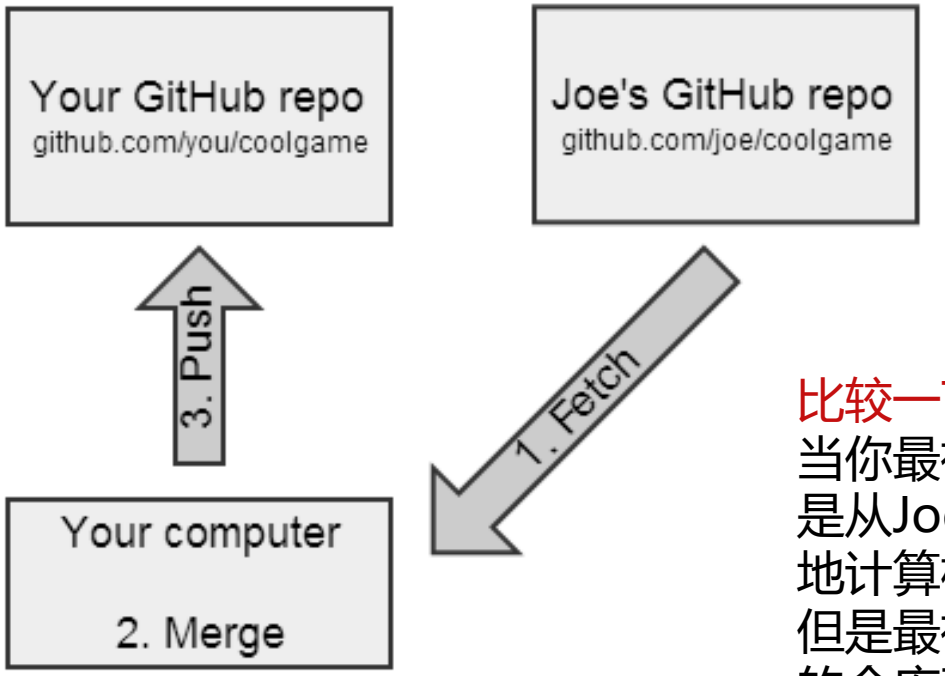
更新某些文件：现在，你可以在任何程序或者环境下更

提交你的更改：这是Git操作。使用该操作让你发送"记
地计算机上完成。

将你的更改push到你的GitHub仓库：这是Git操作。
GitHub。Push操作不会自动完成，所以直到你做了p

6. 给Joe发送一个pull request：如果你认为Joe会接受你

集成管理者——同步一个Fork



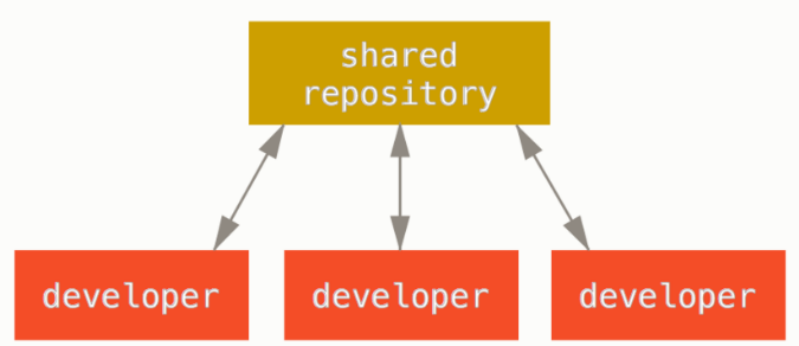
1. 从Joe的仓库中取出那些变化的文件：这;

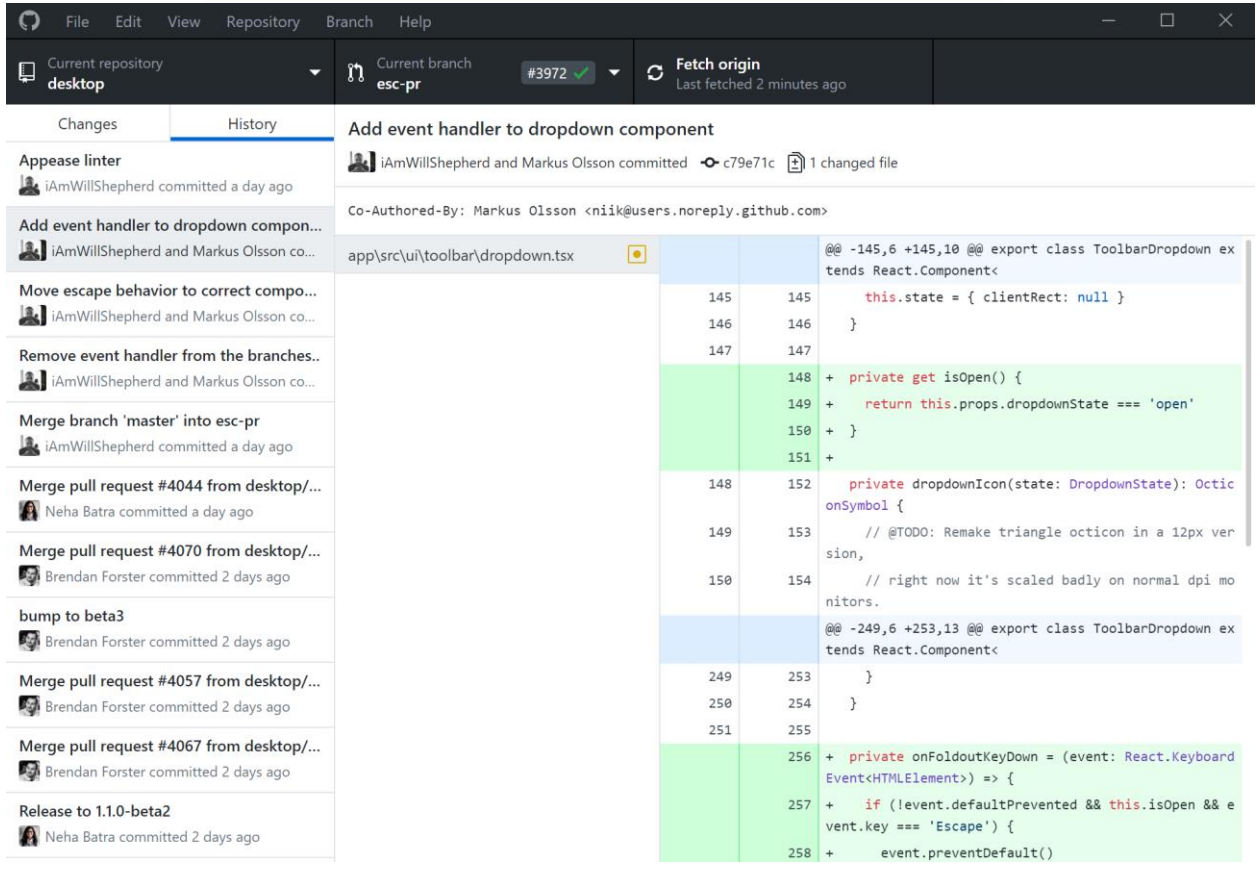
2. 将这些修改合并到你自己的仓库：这是G
   改暂时存放在一个"分支"中）。记住：步

3. 将那些修改更新推送到你的GitHub仓库

比较一下fork和同步工作流程的区别：
当你最初fork一个仓库的时候，信息的流向是从Joe的仓库到你的仓库，然后再到你本地计算机。
但是最初的过程之后，信息的流向是从Joe的仓库到你的本地计算机，之后再到你的仓库。

集中式工作流——常用操作和工具介绍





工具：

1. **GitHub网站和账号**
2. **申请学生认证，无数量限制的私有仓库**
3. **使用GitHub Desktop**

操作：

1. **Fetch origin**
2. **Commit to master**
3. **Push origin**

★ 入门：my-git

★ 进阶：Git Pro中文版

实用技术—Vim/Emacs/VS code/IDE(CCS)

- 简明 Vim 练级攻略
- 一年成为Emacs高手 (像神一样使用编辑器)
- Visual Studio Code入门

通用设置：

1．编码格式：UTF-8

2．缩进：将Tab键设置为2个空格

以CCS为主，其他编辑器做为快速编辑工具和乐趣

正确的编程学习方法

□理论与实践相结合，更注重实践

1. 通过自顶而下的探索与项目实践，获得编程直觉与推动力；
2. 从自底向上的打基础过程中，获得最重要的通用方法并巩固编程思想的理解。

□实践的好处

不同的领域，其“理论/实践”的比例也是不同的。而编程是一个【非常强调实践】的领域。

（1）实践能强化记忆

（2）实践过程中会碰到一些困难（比如碰到 bug），在解决困难的过程中，又能学到新的东西

（3）实践的过程中，还能让你体会到编程的乐趣，称之为“构建的乐趣”。

□编程的乐趣

（1）构建的乐趣：不仅可以模拟世界，更可以创造一个新的世界

（2）在多个抽象层次上工作，抽象层次越高，感觉越纯粹

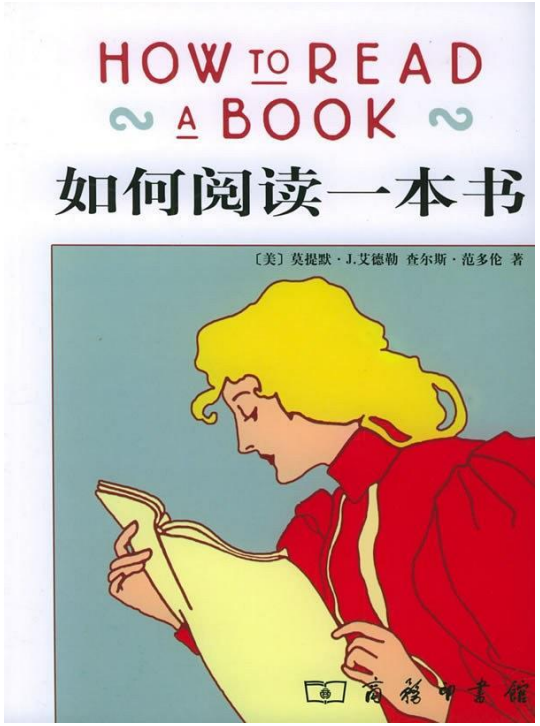
（3）规则的确定性，没有近似，所有问题都有明确的原因

延申：如何自学计算机科学/编程技术

- ★ Teach Yourself Computer Science
- ★ 编程入门指南
- ★ 十年时间学会编程Teach Yourself Programming in Ten Years

第三部分：工作学习方法分享

如何读书——《如何阅读一本书》



- **阅读的目的**：娱乐消遣、获取资讯、增进理解力
- **主动阅读 vs 被动阅读：**带着问题去阅读，要一边阅读一边思考
- **四个基本问题**：
  (1) 整体来说，这本书到底在谈些什么？
  (2) 作者细部说了什么，怎么说的？
  (3) 这本书说得有道理吗？
  (4) 这本书跟你有什么关系？
- **阅读的层次**：
  （1）基础阅读
  （2）检视阅读
  （3）分析阅读
  （4）主题阅读

选书原则：只看经典

（1）看作者

（2）看目录和简介

（3）看亚马逊评价，美亚

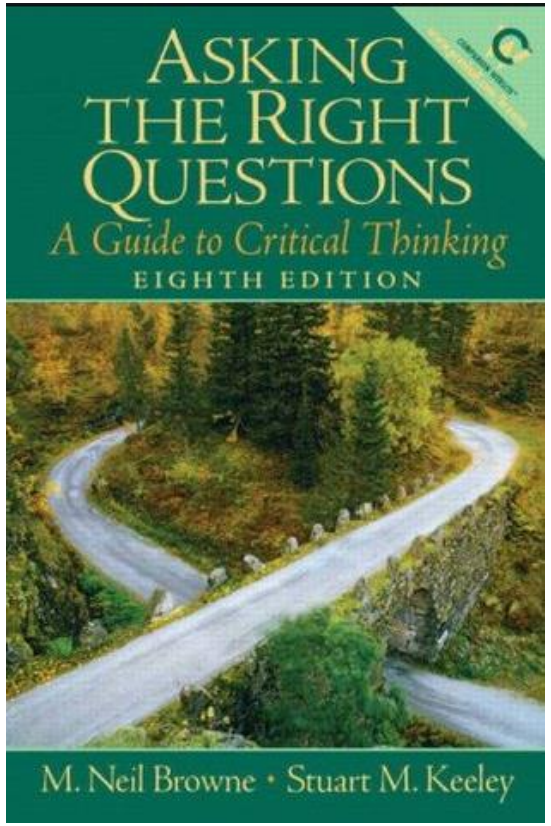
（4）看样章

生命与心智的成长：

*我们的身体是**有限制**的，心智却**没有限制**。其中一个迹象是，在力量与技巧上，身体不能无限制地成长。人们到了30岁左右，身体状况就达到了巅峰，随着时间的变化，身体的状况只有越来越恶化，而**我们的头脑却能无限地成长与发展下去**。*

这是人类最明显的特质，也是万物之灵与其他动物最主要不同之处。**其他的动物似乎发展到某个层次之后，便不再有心智上的发展。但是人类独有的特质，却也潜藏着巨大的危险。**心智就跟肌肉一样，如果不常运用就会萎缩

如何思考——《学会提问-批判性思维指南》



- **海绵式思维和淘金式思维**
- **强势批判思维和弱势批判思维**

2 论题和结论是什么?

3 理由是什么?

4 哪些词句有歧义?

5 什么是价值观冲突和价值观假设?

6 什么是描述性假设?

7 推理过程中有没有谬误?

8 证据的效力如何? (直觉、个人经历、典型案例、当事人证词和专家意见)

9 证据的效力如何? (个人观察、研究报告和类比)

10 有没有替代原因?

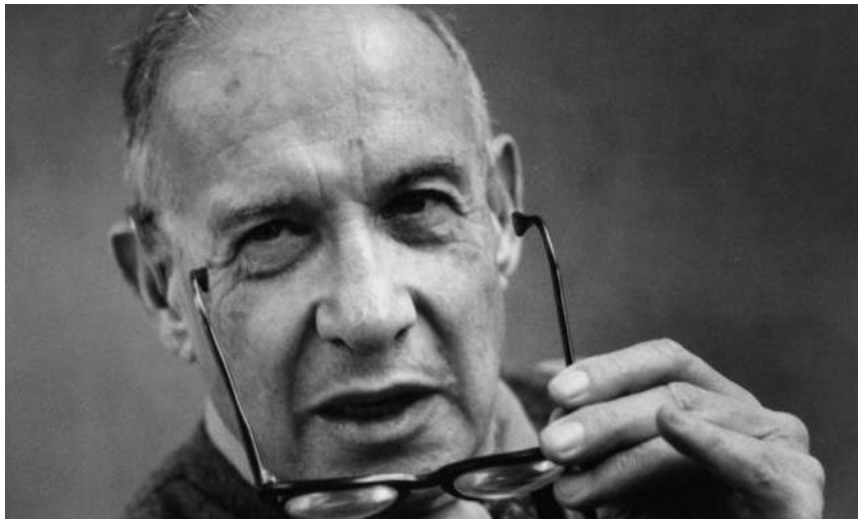
11 数据有没有欺骗性?

12 哪些重要信息被遗漏了?

13 能得出哪些合理的结论?

自我管理——《德鲁克谈自我管理》

★背景介绍
★0、前言
★1、我的长处是什么?
★2、我的工作方式是怎样的?
★3、我如何学习?
★4、我的价值观是什么?
★5、我属于何处?
★6、我该做什么贡献?
★7、我要如何处理人际关系?
★8、我该如何管理后半生?



口我的体会:

(1)战略思维的重要性:工作时间比公司寿命更长,选择比努力更重要

(2)放弃木桶理论:发挥自己的优势,追求卓越

年度计划/时间管理

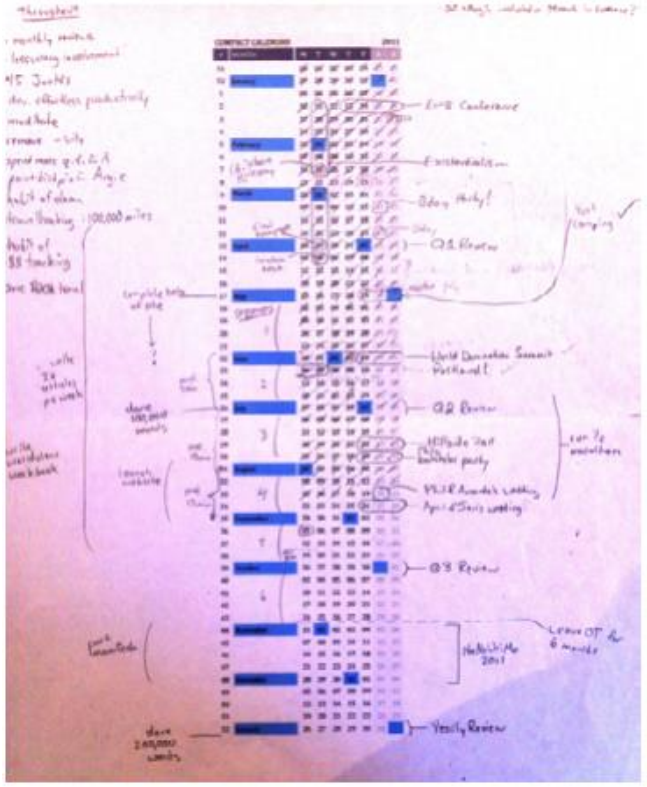


Figure 4: A sample compact calendar.

□ 其他时间管理方法（我尝试过，但不适合）

（1）柳比歇夫时间管理法

（2）把时间当作朋友

（3）剑飞在思考

三电平项目时间分配

求和项:> 持续时间	列标签		
行标签	日常事务	项目执行	总计
软件设计		389:00:48	389:00:48
小样机实验		138:19:36	138:19:36
制造与组装		63:36:45	63:36:45
软件测试		45:41:00	45:41:00
硬件设计		30:14:26	30:14:26
控制器调试		21:19:47	21:19:47
项目进度交流会		18:39:32	18:39:32
采购与合同		17:11:48	17:11:48
小组例会	12:37:23	3:54:42	16:32:05
28377例程学习		15:35:36	15:35:36
整理资料		15:00:16	15:00:16
软硬件初步测试		9:28:18	9:28:18
进度汇报		7:25:54	7:25:54
控制器学习		6:48:31	6:48:31
整理worktile		5:37:13	5:37:13
调研报告		5:14:24	5:14:24
学习提高		5:13:29	5:13:29
28377芯片资料学习		4:26:26	4:26:26
毕设		4:22:05	4:22:05
制定工作计划		4:00:41	4:00:41
采购清单与合同		1:19:01	1:19:01
硬件调试		0:59:38	0:59:38
worktile整理		0:49:51	0:49:51
控制器调试小组例会	0:35:48		0:35:48
总计	13:13:11	814:19:47	827:32:58

备注：从2018.01.08开始统计投入时间，实际从2017.01开始做该项目。有一年时间未进行统计。

养成自学和独立解决问题的习惯

★ 提问的智慧—How To Ask Questions The Smart Way

在你准备要通过电子邮件、新闻群组或者聊天室提出技术问题前，请先做到以下事情：

1. 尝试在你准备提问的论坛的旧文章中搜索答案。
2. 尝试上网搜索以找到答案。
3. 尝试阅读手册以找到答案。
4. 尝试阅读常见问题文件（FAQ）以找到答案。
5. 尝试自己检查或试验以找到答案。
6. 向你身边的强者朋友打听以找到答案。
7. 如果你是程序开发者，请尝试阅读源代码以找到答案。

★ 尽量不打扰别人，心流的重要性—The Joel Test: 12 Steps to Better Code

- We all know that knowledge workers work best by getting into “flow”, also known as being “in the zone”
- The trouble is, getting into “the zone” is not easy
- The other trouble is that it’s so easy to get knocked out of the zone

获取信息的渠道

要超过别人其实还是比较简单的，尤其在今天的中国，更是简单。因为，你只看看中国的互联网，你就会发现，他们基本上全部都是在消费大众，让大众变得更为地愚蠢和傻瓜。**所以，在今天的中国，你基本上不用做什么，只需要不使用中国互联网，你就很自然地超过大多数人了。**当然，如果你还想跟他们彻底拉开，甩他们几个身位，把别人打到底层，下面的这些“技巧”你要多多了解一下。

在信息获取上，你要不断地向大众鼓吹下面的这些事：

- 让大家都用百度搜索引擎查找信息，订阅微信公众号或是到知乎上学习知识……要做到这一步，你就需要把“百度一下”挂在嘴边，然后要经常在群或朋友圈中转发微信公众号的文章，并且转发知乎里的各种“如何看待……”这样的文章，让他们爱上八卦，爱上转发，爱上碎片。
- 让大家到微博或是知识星球上粉一些大咖，密切关注他们的言论和动向……是的，告诉大家，大咖的任何想法一言一行都可以在微博、朋友圈或是知识星球上获得，让大家相信，你的成长和大咖的见闻和闲扯非常有关系，你跟牛人在一个圈子里你也会变牛。

★ 如何超过大多数人

口如何跨越新的长城

（1）github-lantern

（2）protonvpn

强烈推荐——博客program-think（谷歌搜索）

3个重要部分

口博客文章和评论区

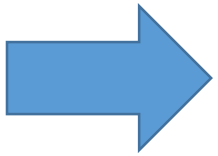
口电子书

口收藏的网站

★ **隐私保护工具**

关键问题

- **你想成为什么样的人/过什么样的生活?**
- **你的优势和兴趣是什么?**
- **从事什么工作/行业?**



未来2-3年怎样度过

问题?

祝大家学有所成