Technology Tarena 达内科技

Unix系统高级编程

开发环境 Unit01

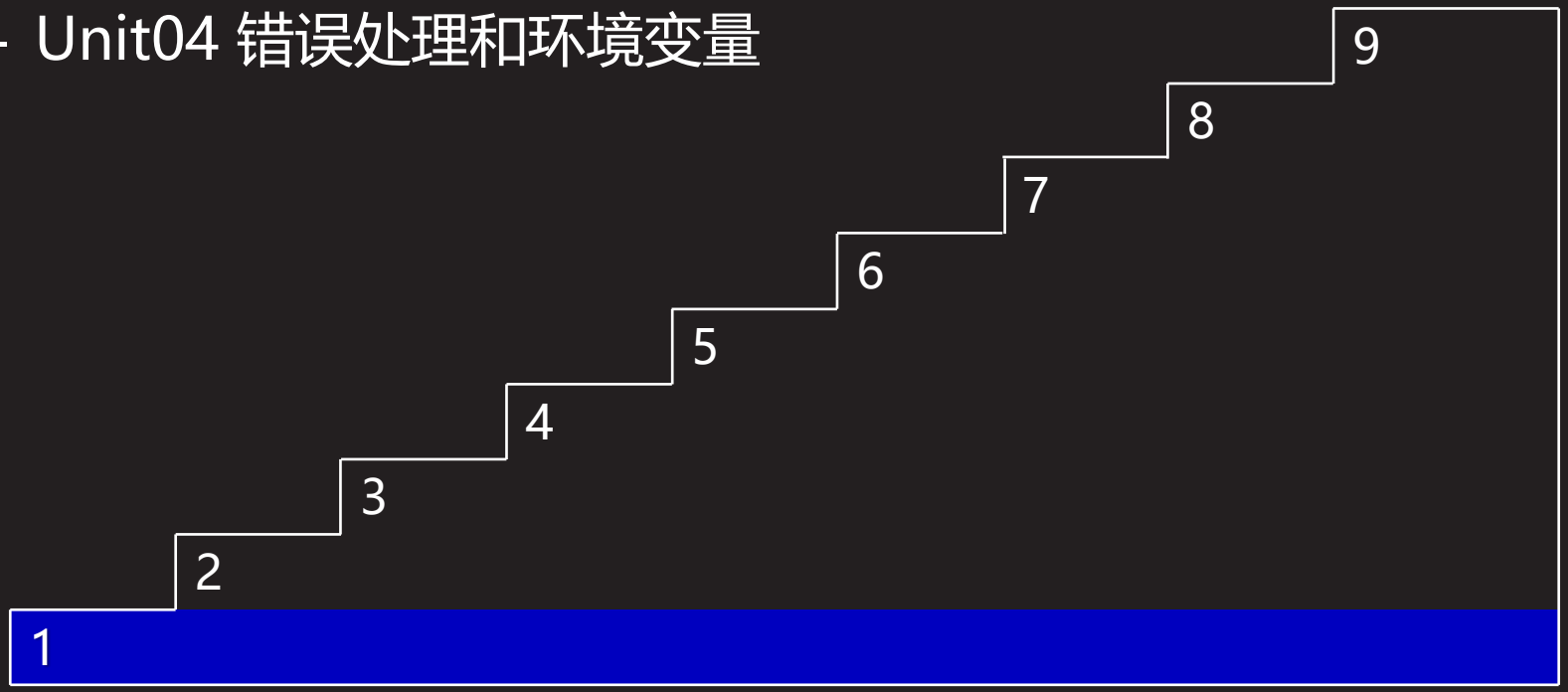
Technology Tarena 达内科技

课程介绍

课程介绍

- Part 1 起步
  - Unit01 开发环境
  - Unit02 静态库和共享库
  - Unit03 动态加载和辅助工具
  - Unit04 错误处理和环境变量

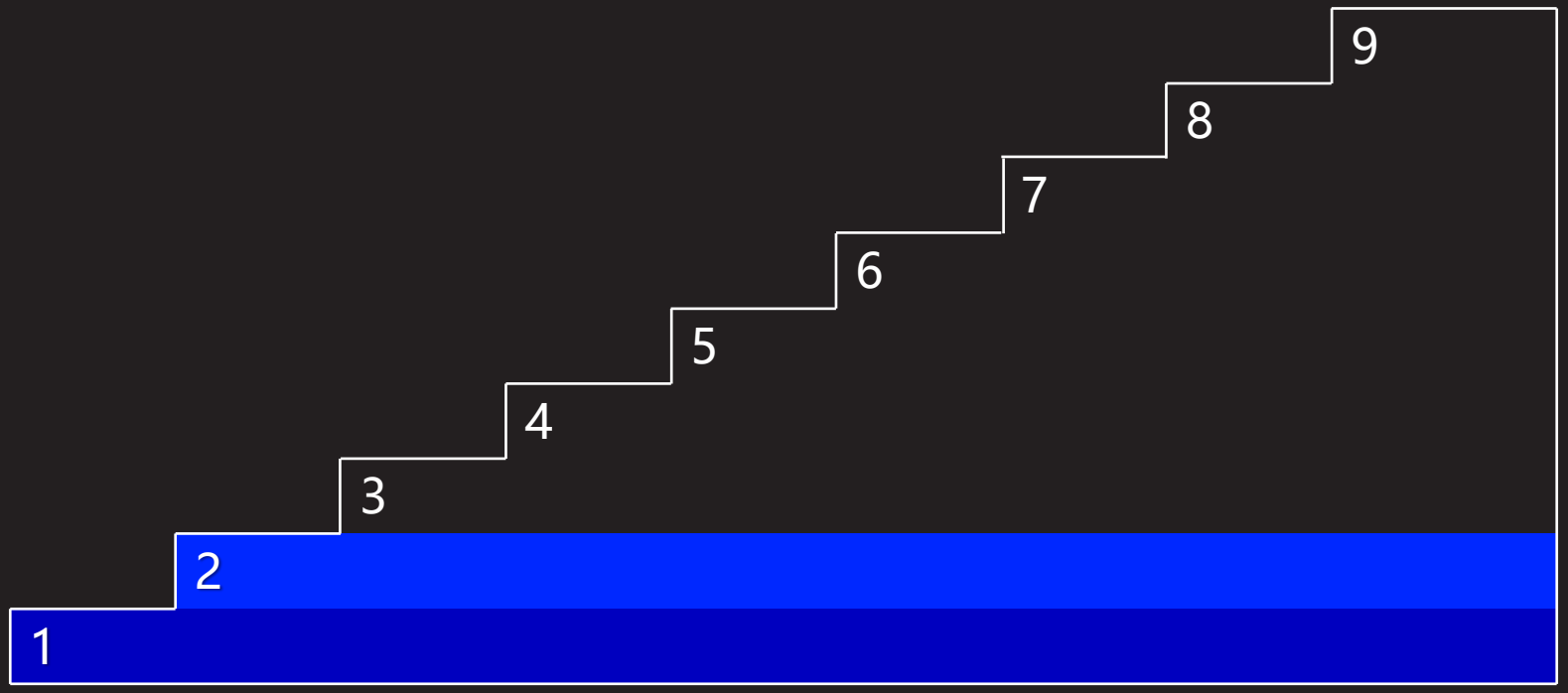
知识讲解



课程介绍（续1）

- Part 2 内存
  - Unit05 进程映像和虚拟内存
  - Unit06 内存的分配与释放

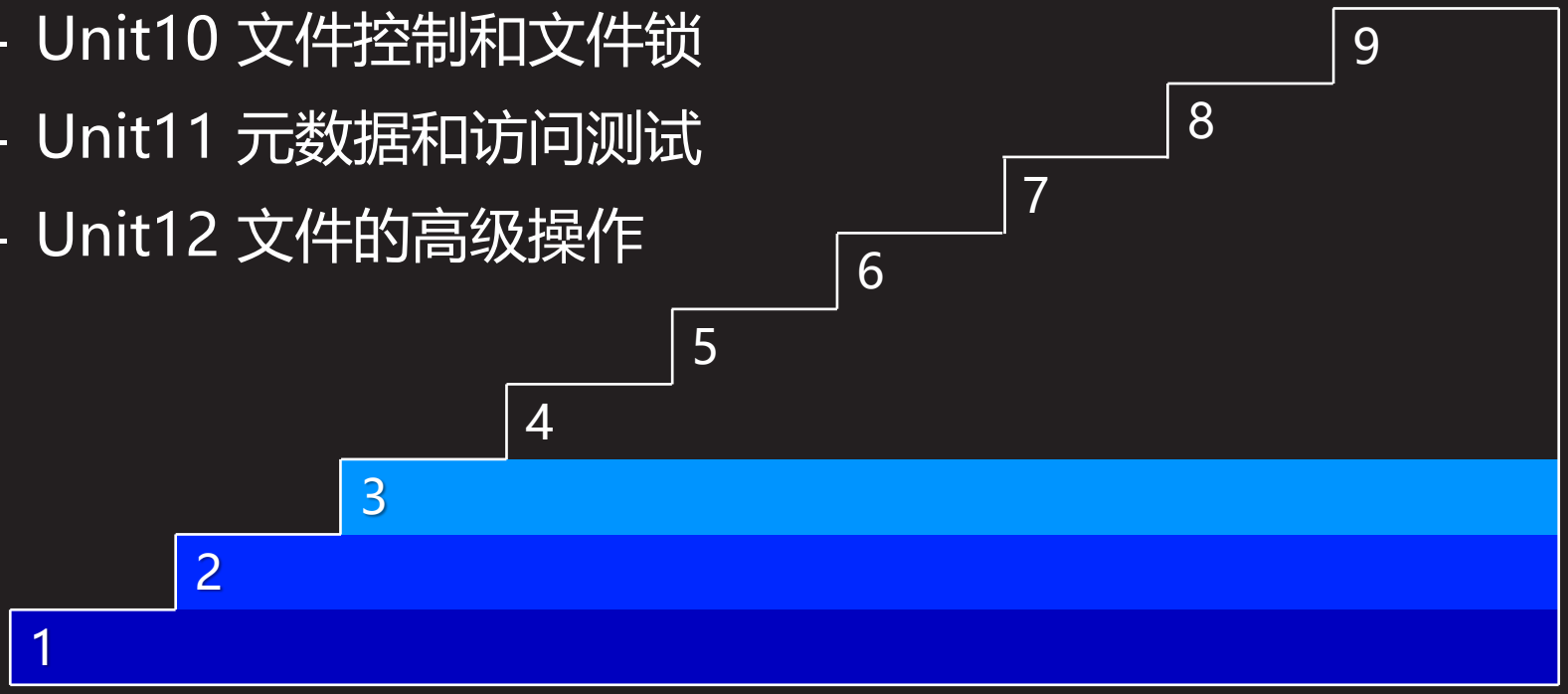
知识讲解



课程介绍（续2）

- Part 3 文件
  - Unit07 系统调用和文件系统
  - Unit08 文件的基本操作
  - Unit09 随机访问和文件同步
  - Unit10 文件控制和文件锁
  - Unit11 元数据和访问测试
  - Unit12 文件的高级操作

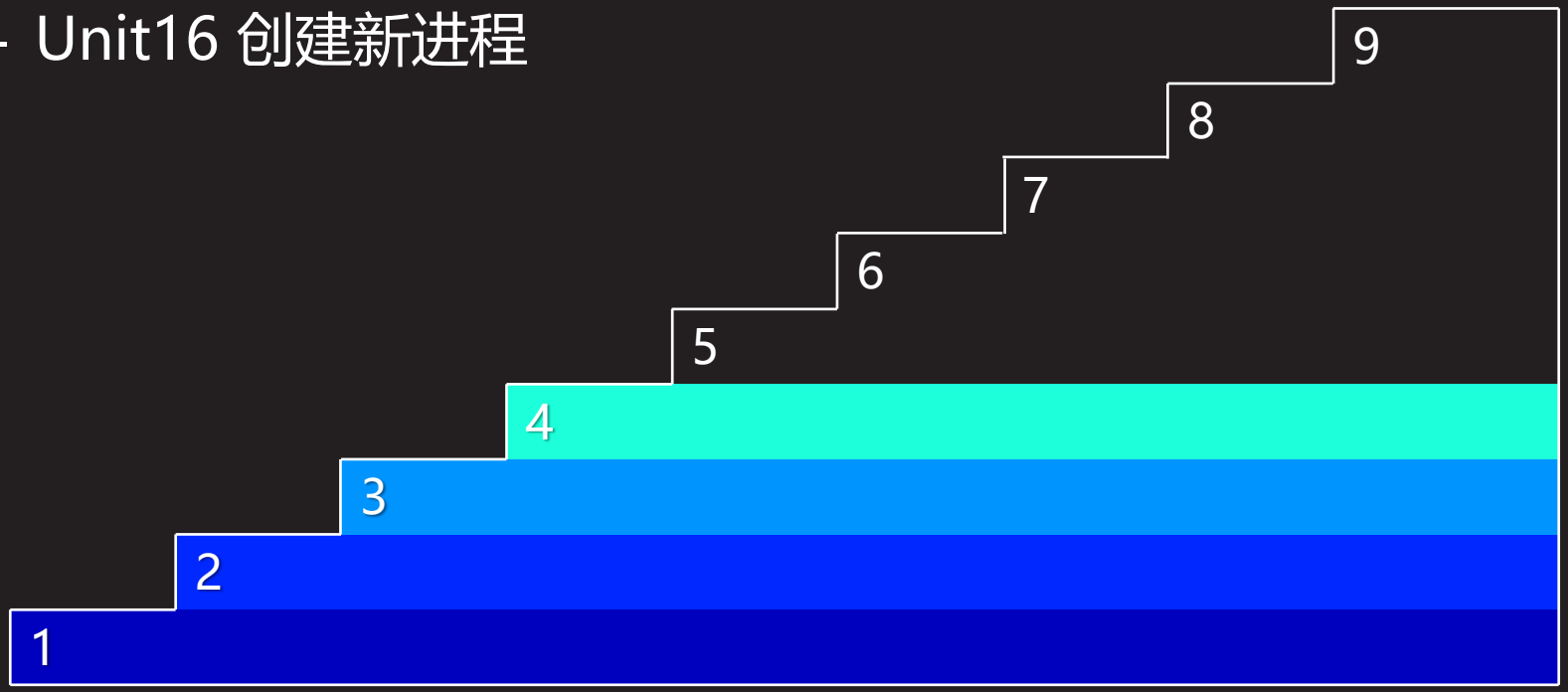
知识讲解



课程介绍（续3）

- Part 4 进程
  - Unit13 进程和进程ID
  - Unit14 创建子进程
  - Unit15 进程的终止与回收
  - Unit16 创建新进程

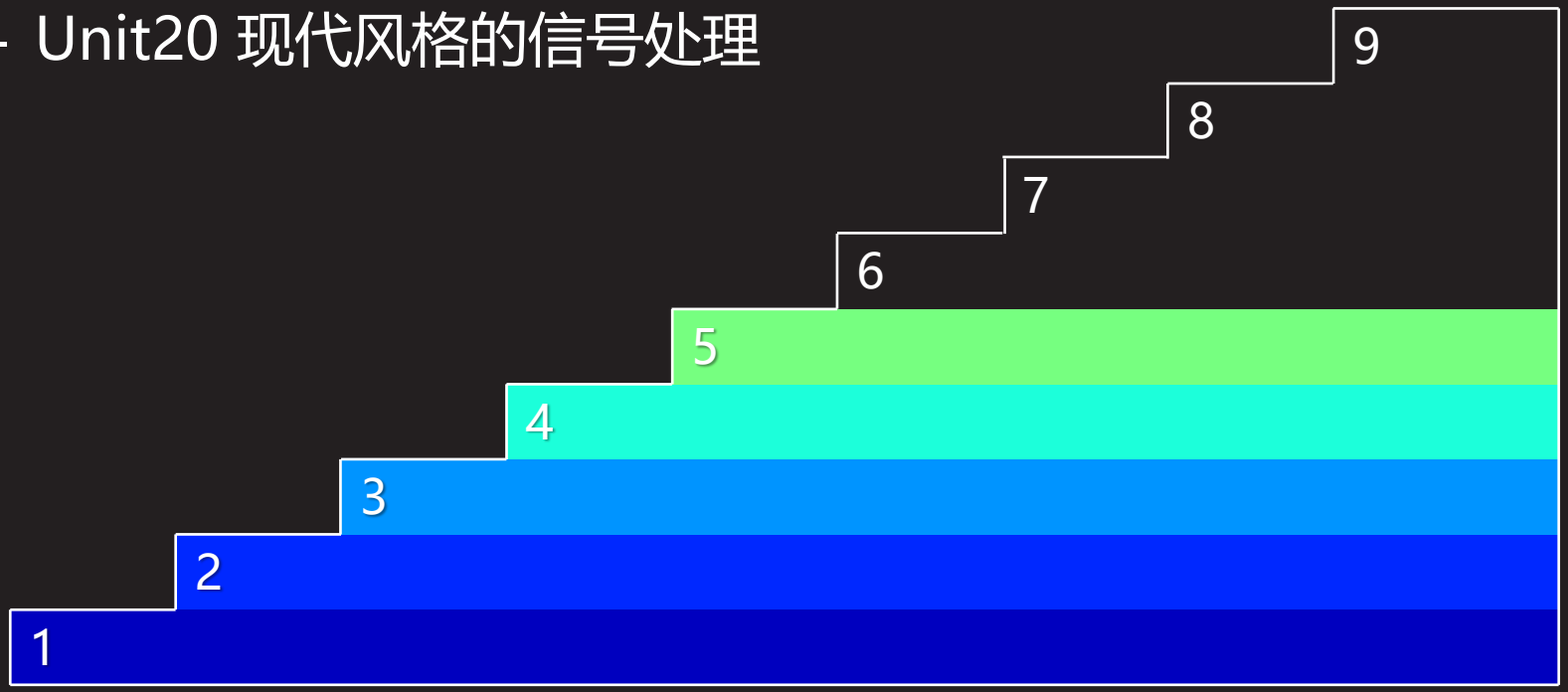
知识讲解



课程介绍（续4）

- Part 5 信号
  - Unit17 信号和信号捕获
  - Unit18 信号的发送与定时
  - Unit19 信号屏蔽和定时器
  - Unit20 现代风格的信号处理

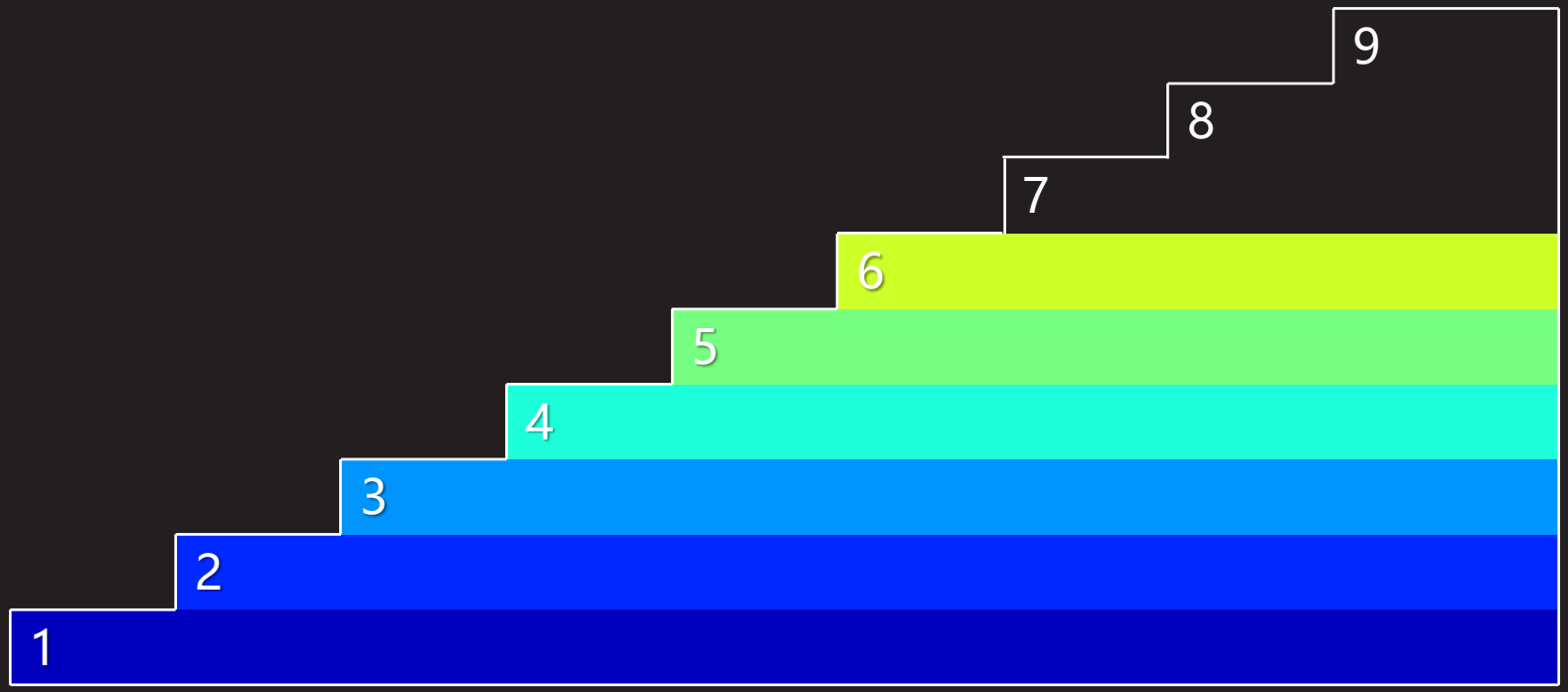
知识讲解



课程介绍（续5）

- Part 6 进程间通信
  - Unit21 进程间通信和管道
  - Unit22 XSI的IPC对象

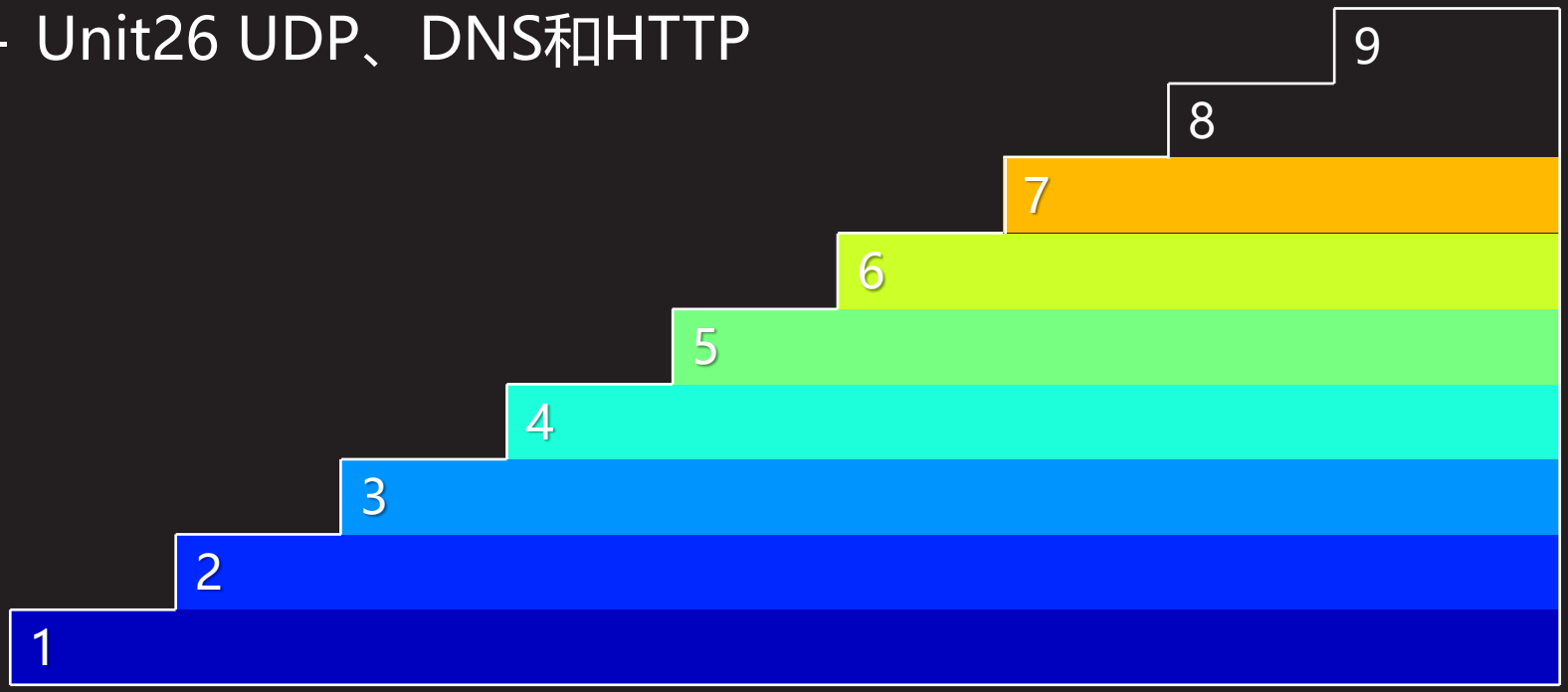
知识讲解



课程介绍（续6）

- Part 7 网络
  - Unit23 网络和网络协议
  - Unit24 套接字编程
  - Unit25 TCP客户机/服务器
  - Unit26 UDP、DNS和HTTP

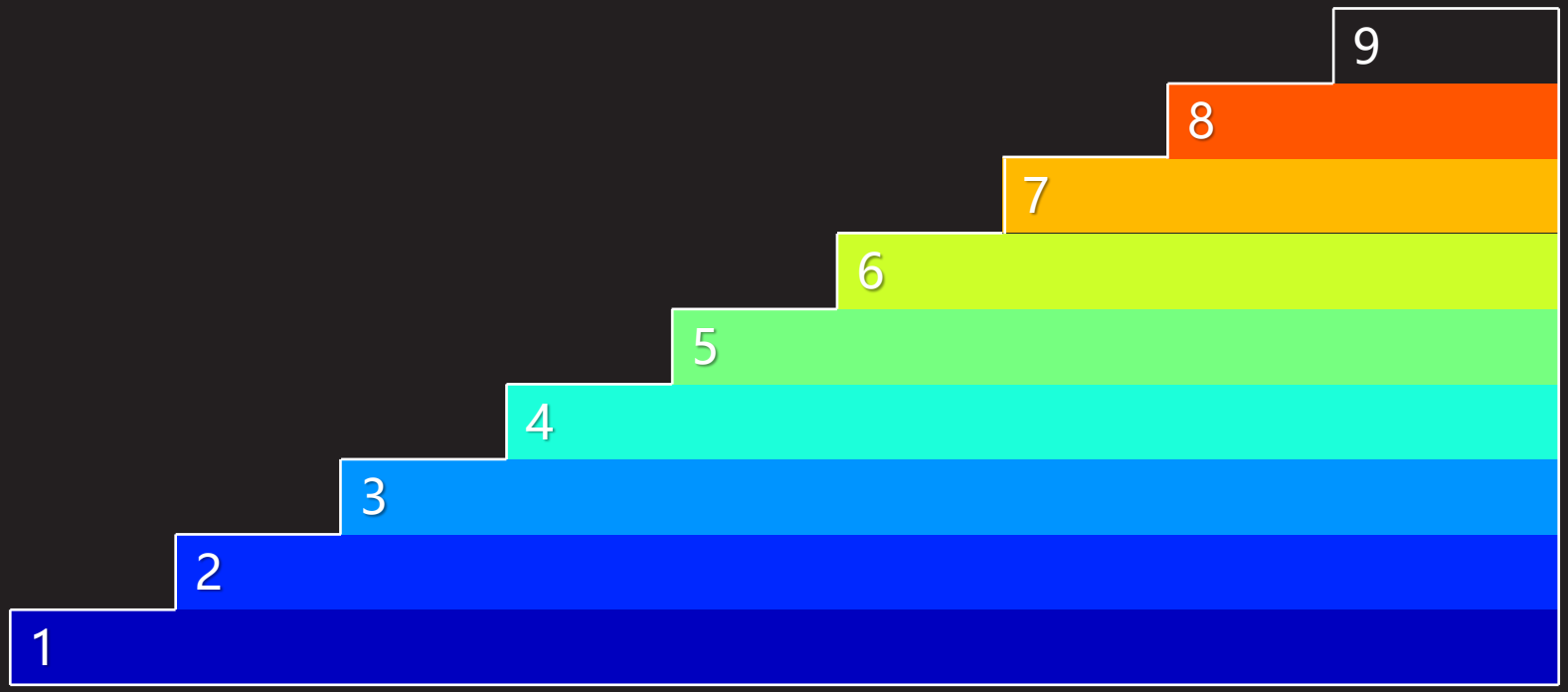
知识讲解



课程介绍（续7）

- Part 8 线程
  - Unit27 线程和线程控制
  - Unit28 并发冲突和线程同步

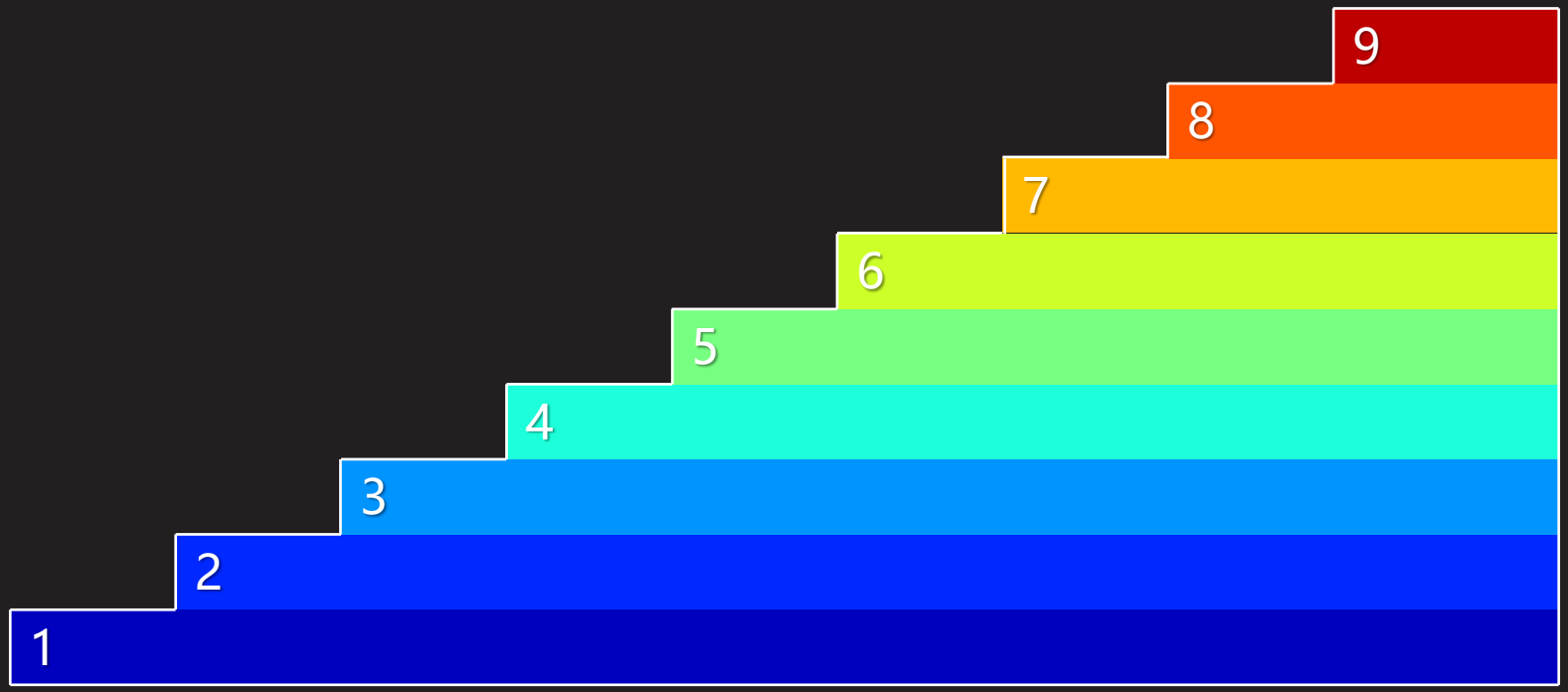
知识讲解



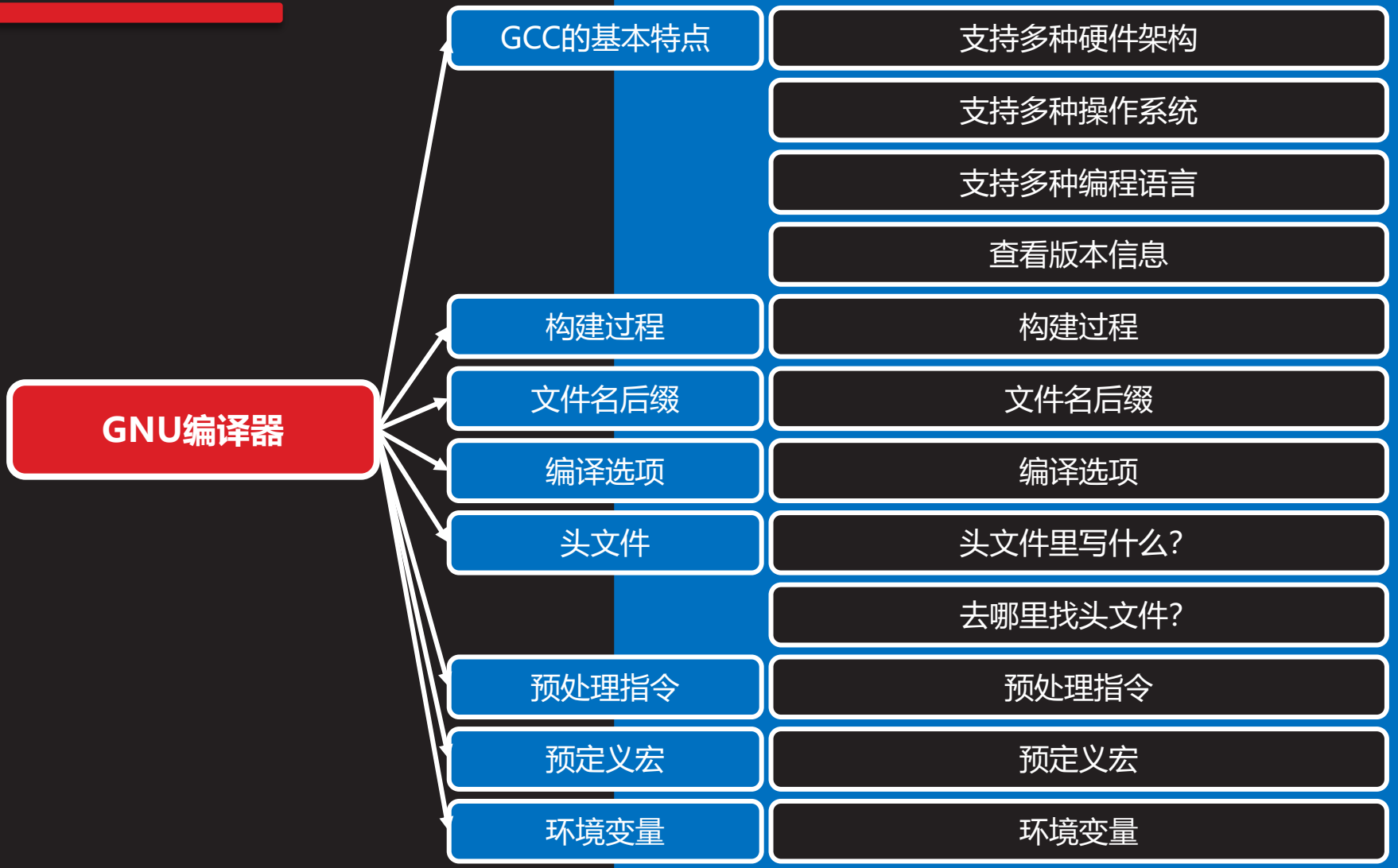
课程介绍（续8）

- Part 9 项目
  - Unit29 Web服务器项目(一)
  - Unit30 Web服务器项目(二)

知识讲解



GNU编译器

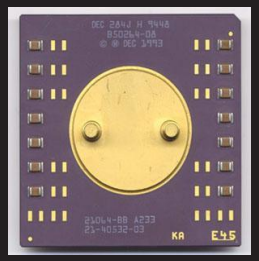

GNU编译器
GCC的基本特点
支持多种硬件架构
支持多种操作系统
支持多种编程语言
查看版本信息
构建过程
构建过程
文件名后缀
文件名后缀
编译选项
编译选项
头文件
头文件里写什么?
去哪里找头文件?
预处理指令
预处理指令
预定义宏
预定义宏
环境变量
环境变量

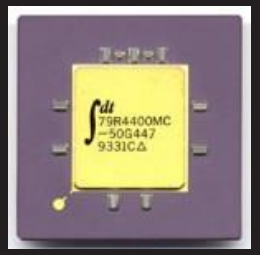
Technology Tarena 达内科技

GCC的基本特点

支持多种硬件架构

- x86-64
- Alpha
- ARM
- Motorola 68000
- MIPS
- PDP-10/11
- PowerPC
- System/370-390
- SPARC
- VAX

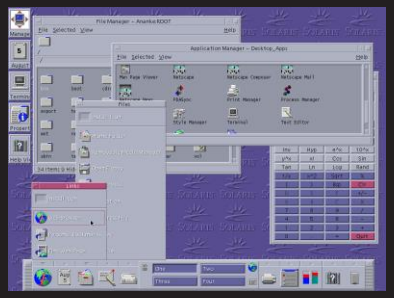


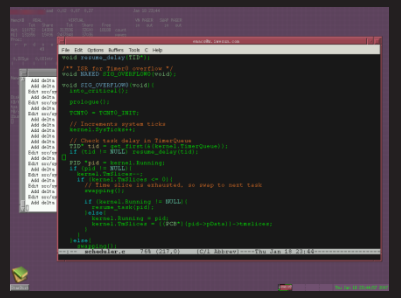


知识讲解

支持多种操作系统

- Unix
- Linux
- BSD
- Android
- Mac OS X
- iOS
- Windows





支持多种编程语言

- C
  - 以简洁高效的方式控制系统底层，无需环境支持便能运行
- C++
  - 兼容C语言，集面向过程、面向对象和泛型编程于一身
- Objective-C
  - 扩充C语言的面向对象特性，主要用于Mac OS X和GNUstep这两个使用OpenStep标准的系统
- Java
  - 面向对象且跨平台， 需要虚拟机支持，号称一次编译到处运行

支持多种编程语言（续1）

- Fortran
  - 最早的高级程序设计语言，主要用于科学和工程计算
- Pascal
  - 最早的结构化程序设计语言，语法严谨，层次分明，擅于描述算法和数据结构
- Ada
  - 迄今为止最复杂最完备的程序设计语言，代表了全世界程序设计语言领域的最高成就，甚至突破了冯诺依曼思维模式的桎梏，标志着软件工程已经发展到国家和国际的规模
  - 美国国防部指定的唯一一种可应用于军事系统开发的语言
  - 我国军方也将其作为军内系统开发标准：GJB1383-1998

知识讲解

查看版本信息

- GCC的早期版本只能编译C语言程序
  - GNU C Compiler
- GCC现在的版本已可以编译多种语言程序
  - GNU Compiler Collection
- 查看编译器版本

```
$ gcc -v
```

知识讲解

Technology Tarena 达内科技

构建过程

构建过程

- 从源代码到可执行程序的构建过程
  1. 预编译(编译预处理)：头文件扩展、宏扩展
     $ gcc -E hello.c -o hello.i -> hello.i
  2. 编译：将高级语言翻译成汇编语言，得到汇编文件
     $ gcc -S hello.i -> hello.s
  3. 汇编：将汇编语言翻译成机器指令，得到目标模块
     $ gcc -c hello.s -> hello.o
  4. 链接：将目标模块和标准库链接，得到可执行程序
     $ gcc hello.o -o hello -> hello
- 运行

```
$ hello
```

知识讲解

Hello, World !

【参见：hello.c】

课堂练习

- Hello, World !

Technology Tarena 达内科技

文件名后缀

文件名后缀

- `.h` ：C语言源代码头文件
- `.c` ：预处理前的C语言源代码文件
- `.i` ：预处理后的C语言源代码文件
- `.s` ：汇编语言文件
- `.o` ：目标文件
- `.a` ：静态库文件
- `.so`：共享库(动态库)文件

知识讲解

Technology Tarena 达内科技

编译选项

编译选项

$ gcc [选项] [参数] 文件1 文件2 ...

知识讲解

- `-o`：指定输出文件
  - $ gcc hello.c -o hello
- `-E`：预编译，缺省输出到屏幕，用-o指定输出文件
  - $ gcc -E hello.c -o hello.i
- `-S`：编译，将高级语言文件编译成汇编语言文件
  - $ gcc -S hello.c
- `-c`：汇编，将汇编语言文件汇编成机器语言文件
  - $ gcc -c hello.c

编译选项（续1）

- `-Wall` ：产生尽可能多的警告
  - $ gcc -Wall wall.c
- `-Werror` ：将警告作为错误处理
  - $ gcc -Werror werror.c
- `-x` ：显式指定源代码的语言
  - $ gcc -x c++ cpp.c -lstdc++
- `-pedantic` ：对非标准的扩展语法产生警告
- `-g` ：产生调试信息
- `-O0/O1/O2/O3` ：指定优化等级，O0不优化，缺省O1

知识讲解

产生尽可能多的警告

【参见：wall.c】

课堂练习

- 产生尽可能多的警告

将警告作为错误处理

【参见：werror.c】

课堂练习

- 将警告作为错误处理

指定源代码的语言

【参见：cpp.c】

课堂练习

- 指定源代码的语言

Technology Tarena 达内科技

头文件

头文件里写什么?

- 头文件卫士
  - `#ifndef __GEOMETRY_H__`
    `#define __GEOMETRY_H__`
    `...`
    `#endif`
- 包含其它头文件
  - #include <sys/types.h>
- 宏定义
  - #define PI 3.14159
- 自定义类型
  - struct Circle { double x; double y; double r; };

知识讲解

头文件里写什么？（续1）

- 类型别名
  - typedef struct Circle CIRCLE;
- 外部变量声明
  - extern double e /* = 2.71828 */;
- 函数声明
  - double circleArea (CIRCLE const*);
- 函数定义不要写在头文件里
  - 一个头文件可能会被多个源文件包含，写在头文件里的函数定义也会因此被预处理器扩展到多个包含该头文件的源文件中，并在编译阶段被编译到多个不同的目标文件中，这将直接导致链接错误：multiple definition，多重定义

去哪里找头文件?

- 通过gcc的-I选项指定头文件的附加搜索路径
  - gcc math.c calc.c -I.
- #include <calc.h>
  - 先找-I指定的目录，再找系统目录
- #include "calc.h"
  - 先找-I指定的目录，再找当前目录，最后找系统目录
- 头文件的系统目录
  - /usr/include
  - /usr/local/include
  - /usr/lib/gcc/i686-linux-gnu/<版本号>/include
  - /usr/include/c++/<版本号> (C++编译器)

知识讲解

分文件声明、定义和调用函数

【参见：calc.h、calc.c、math.c】

课堂练习

- 分文件声明、定义和调用函数

Technology Tarena 达内科技

预处理指令

预处理指令

- `#include` ：将指定文件的内容插至此指令处
- `#define` ：定义宏
- `#undef` ：删除宏
- `#if` ：如果
- `#ifdef` ：如果宏已定义
- `#ifndef` ：如果宏未定义
- `#else` ：否则，与#if/#ifdef/#ifndef配合使用
- `#elif` ：否则如果，与#if/#ifdef/#ifndef配合使用
- `#endif` ：结束判定，与#if/#ifdef/#ifndef配合使用

知识讲解

预处理指令（续1）

- **#error** ：产生错误，结束预处理
  - #error "版本太低！"
- **#warning**：产生警告，不结束预处理
  - #warning "版本太高！"
- **#line** ：指定行号
  - #line 100
- **#pragma** ：设定编译器的状态或指示编译器的操作
  - #pragma GCC dependency "dep.c"
  - #pragma GCC poison goto
  - #pragma pack(1)

知识讲解

产生错误和警告

【参见：error.c】

课堂练习

- 产生错误和警告

指定行号

【参见：line.c】

课堂练习

- 指定行号

编译器指示

【参见：pragma.c、dep.c】

课堂练习

- 编译器指示

Technology Tarena 达内科技

预定义宏

预定义宏

- `__BASE_FILE__` ：正在编译的源文件名
- `__FILE__` ：所在文件名
- `__LINE__` ：行号
- `__FUNCTION__` ：函数名
- `__func__` ：同__FUNCTION__
- `__DATE__` ：日期
- `__TIME__` ：时间
- `__INCLUDE_LEVEL__` ：包含层数，从0开始
- `__cplusplus` ：C++有定义，C无定义

知识讲解

打印预定义宏

【参见：print.h、predef.h、predef.c】

课堂练习

- 打印预定义宏

Technology Tarena 达内科技

环境变量

环境变量

- C_INCLUDE_PATH
  - C头文件的附加搜索路径，相当于gcc的-I选项
- CPATH
  - 同C_INCLUDE_PATH
- CPLUS_INCLUDE_PATH
  - C++头文件的附加搜索路径
- LIBRARY_PATH
  - 链接时查找静态库和共享库的路径
- LD_LIBRARY_PATH
  - 运行时查找共享库的路径

知识讲解

环境变量（续1）

Technology Tarena 达内科技

- 举例
  - 通过-I选项将当前目录作为C头文件附加搜索路径

```
$ gcc calc.c cpath.c -I.
```

  - 将当前目录添加到CPATH环境变量中

```
$ export CPATH=$CPATH:.
```

  - 将对CPATH环境变量的设置写到登录脚本中

```
$ vi ~/.bashrc
```

或

```
$ vi ~/.bash_profile
```

写入：export CPATH=$CPATH:.

知识讲解

环境变量（续2）

- 举例
  - 重新登录，或手动执行登录脚本

```
$ source ~/.bashrc
```

或

```
$ source ~/.bash_profile
```

  - 在登录脚本中设置环境变量，每次登录都会自动生效

环境变量（续3）

- 头文件的三种定位方式
  - #include "<目录>/xxx.h"
    头文件路径发生变化，必须修改源程序
  - export CPATH=$CPATH:<目录>
    同时构建多个工程，容易引发冲突
  - gcc -I<目录>
    既不需要修改源程序，也不会有任何冲突，推荐使用此方法

知识讲解

头文件的附加搜索路径

【参见：calc.h、calc.c、cpath.c】

课堂练习

- 头文件的附加搜索路径

Technology Tarena 达内科技

总结和答疑