



CHINA
PyCon 2011

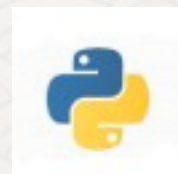
与 Python 一路走来

2011.12.04

初遇 Python



- 2004 年，几年的 C/C++ 磨练，准备学 Java
- 在 china-pub 买书的时候，不知道为啥买了一本《Python 编程宝典》
- 被 Python 缩进风格吸引，丢掉《Java 编程思想》
- 这本书后来送给了一位 pyer(西安 ?)
- 很讨厌早期的图标，我怕蛇



国内第一本 Python 中文书



· [单击此处编辑母版文本样式](#)

- 第二级

- 第三级

- 第四级

- 第五级



涉世之初的迷茫



- 没有顺手的 IDE 目前还是 UliPad 顺手
- 源码中出现中文，程序无法运行 文件头使用 UTF-8 标志
- 有问题没人帮忙 找到 python.cn 邮件列表，跟随木头转战 Google-group
- 不知道可以用来写什么程序
 - Tk 太难看了，习惯 C++ Builder 的我接受不了
 - 当时基本不用 Linux，没啥脚本需求
 - 用 PSP 写 web，真的不想这么干
 - 周围的人都劝我放弃吧，没前途的



该不该放弃呢？

加盟新公司

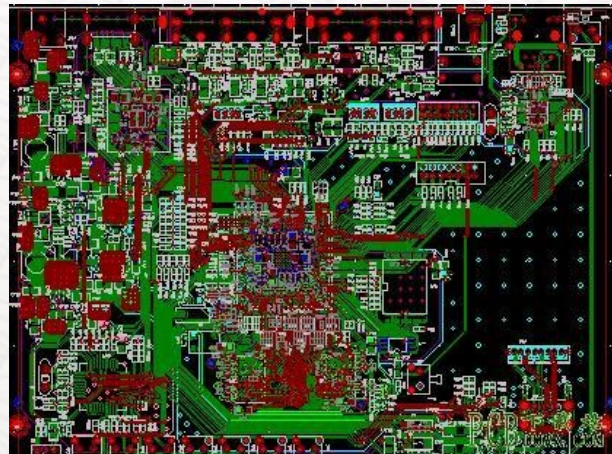


- 2005 年的 9 月，加盟了一家创业公司
- 该公司主要从事电子制造业软件的开发
- 涉及到上百种设备的数据需要处理
- 数据格式有文本和二进制的
- 厂家不公开格式，需要自己 hack
- 配套的工具种类繁多
- 负责新版软件的架构设计





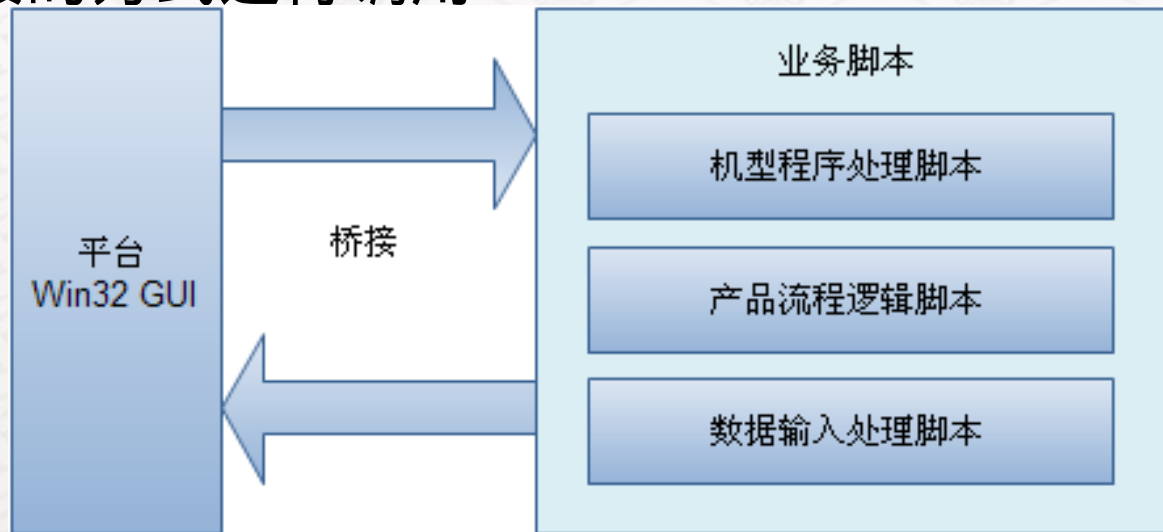
- 定位于桌面客户端
- 机型复杂，不同客户的机型配置不一样
- 设备程序的优化要考虑用户生产的机种
- 输入的数据格式至少有好几百种
- 现有的架构是一堆的 exe+dll
- 交互频繁，矢量画图，数据量大
- 响应客户需求变化的时间少则几天，多则...



新架构的遐想



- 借鉴 HW 公司的架构模式：稳定的平台 + 灵活的业务
- 平台专注于人机交互这块
- 业务专注于机型和流程控制
- 平台和业务通过桥接的方式进行调用





- 当时知道的脚本： csh、 perl、 python
- Python 是胶水语言
- Python 有 C 的 API 可以调用，嵌入到主程序中
- 受 HW 的平台影响很深，业务一定得是脚本

最后选择 Python



- 帮助文档有 C API 的详细介绍
- 没有任何的中文资料
- 没有详细的例子可以参考
- 基本无法调试
- 引用计数器的痛苦

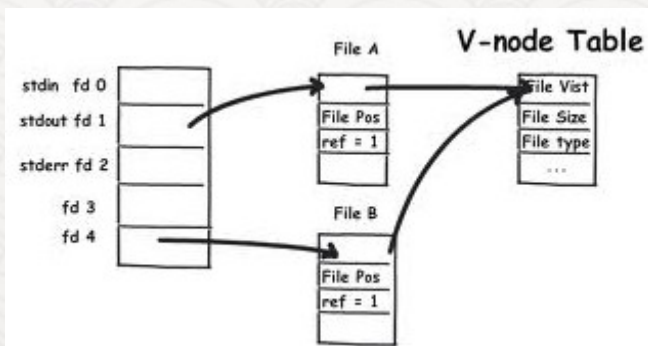


python272.chm

Google

```
>>> import this
The Zen of Python, by Tim Peters
```

```
Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
```





- 捕获并重定向

- stdout
- stderr

- 区分版本

- Debug
- Release

```
PyObject* log_CaptureStdout(PyObject* self, PyObject* pArgs)
{
    char* LogStr = NULL;
    if (!PyArg_ParseTuple(pArgs, "s", &LogStr)) return NULL;

    #ifdef _DEBUG
    logFile << LogStr;
    std::flush(logFile);
    #endif

    Py_INCREF(Py_None);
    return Py_None;
}

//-----
PyObject* log_CaptureStderr(PyObject* self, PyObject* pArgs)
{
    char* LogStr = NULL;
    if (!PyArg_ParseTuple(pArgs, "s", &LogStr)) return NULL;

    logFile << LogStr;
    std::flush(logFile);

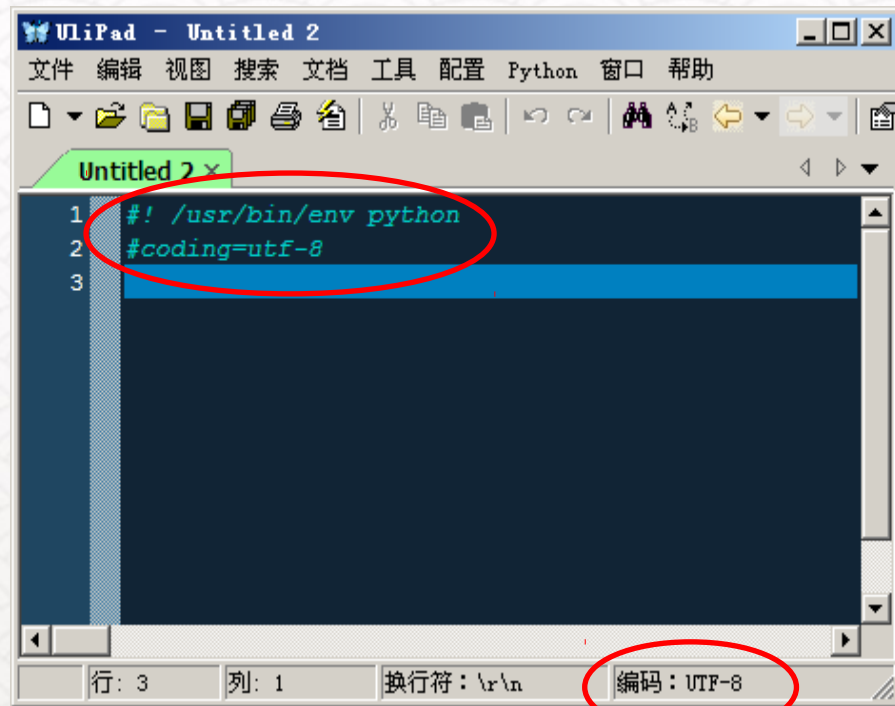
    Py_INCREF(Py_None);
    return Py_None;
}

//-----
static PyMethodDef logMethods[] = {
    {"CaptureStdout", log_CaptureStdout, METH_VARARGS, "Logs stdout"},
    {"CaptureStderr", log_CaptureStderr, METH_VARARGS, "Logs stderr"},
    {NULL, NULL, 0, NULL}
};
```

逐个击破 - 中文处理



- 文件首行: #coding:utf-8
- 源代码保存为 UTF-8 编码
- 对中文目录进行 Unicode 处理
- 对数据文件进行编码处理
 - 有些文件需要转为本地编码
 - 有些文件需要 UTF-8 编码
 - 无法进行编码转换的字符处理
- 工控程序中的中文替换处理



数据分析处理

遇到的问题

- 工控程序两种格式
 - 文本
 - 二进制
- 程序格式厂家没有公开的，需要自己分析
- 每种机型的格式都是不一样的

处理结果

- 平台统一接口调用脚本
- 脚本以模块 / 类继承的方式组织
- 数据传递
 - 接口
 - 文件
 - 数据库

动态调用



Owner 为厂家名称

device 为厂家下的设备

```
# 调用指定目录下的 pana/bm123.py
```

```
mod = loadModule("pana","bm123")
```

```
obj = mod.Smt()
```

```
obj.saveProg()
```

```
def loadModule(owner, device):
    logger.info('smt.LoadModule(%s,%s)'%(owner, device))
    name = 'smt.%s.%s'%(owner.lower(), device.replace('-', '').lower())
    logger.info('Module:%s'%name)
    module = None
    #机型
    logger.info('Load DEVICE module:%s...'%name)
    try:
        module = __import__(name)
    except Exception, msg:
        logger.info(msg.__str__())
        #品牌
        name = '.'.join(name.split('.')[:-1])
        logger.info('Load OWNER module:%s...'%name)
        try:
            module = __import__(name)
        except Exception, msg:
            logger.info(msg.__str__())

    if module is not None:
        components = name.split('.')
        for comp in components[1:]:module = getattr(module, comp)#获取模块
    logger.info(module)
    return module
```



- 绘图文件的指令经常几十万行，甚至上百万行
- 读取操作在脚本中，无法在界面显示进度
- 相比 C 写的，速度太慢，基本都要几秒钟，甚至需要几十秒
- 使用 psycos ，脚本执行 OK ，但平台调用接口会出错
- 使用 map 来代替 for 循环，速度有提升

不要迷恋语言性能，业务性能才是关键！



· 内部

- 文本操作从5000多行降到800行，轻松维护和使用
- 去除了大量的.dll，换来了更易维护的.py
- 目录结构就是类的结构，代码条理性更强
- 极大的降低了研发的工作量

· 外部

- 支持相同的机型(100多种)，我们用了2年时间，竞争对手用了7-8年
- 很多需求我们可以做到现场解决，而竞争对手需要3-10天(数据需要现场采集)
- 现场解决只需要售前工程师，而对手需要售前+研发等

软件 SaaS 化



- 单机版盗版严重
- 获取用户的运行数据困难
- 低端竞争对手风起云涌
- 工厂网速在提升和开放

步序	料站	元件名	X	Y	角度	料号	外形码	类型	站宽	速度	优先级	吸嘴	拼板	描述
1	17	AR11	25.044	133.934	0	10G212472004030:2		Tape				-1		RES 4.7K 0HM 1/16W (0402) 5%QVL: 國巨
2	17	AR457	22.504	98.12	90	10G212472004030:2		Tape				-1		RES 4.7K 0HM 1/16W (0402) 5%QVL: 國巨
3	19	C657	105.816	200.99	0	11G232110211310:1		Tape				-1		MLCC 1000PF/16V (0402)X7R10% TDK/C10...
4	19	C252	105.816	202.133	0	11G232110211310:1		Tape				-1		MLCC 1000PF/16V (0402)X7R10% TDK/C10...
5	21	R1530	39.116	46.786	90	10G212000004030		Tape				-1		RES 0 0HM1/16W (0402)JUMPTAGED/BC0402...
6	21	AR55	9.042	117.932	0	10G212000004030		Tape				-1		RES 0 0HM1/16W (0402)JUMPTAGED/BC0402...
7	23	C253	108.483	269.062	135	11G232110211310:2		Tape				-1		MLCC 1000PF/16V (0402)X7R10% TDK/C10...
8	23	C254	110.007	270.84	135	11G232110211310:2		Tape				-1		MLCC 1000PF/16V (0402)X7R10% TDK/C10...
9	25	AR21	6.756	108.661	135	10G212473004010:1		Tape				-1		RES 47K 0HM 1/16W (0402) 5%TA-I/EM04...
10	25	AR20	6.731	106.502	0	10G212473004010:1		Tape				-1		RES 47K 0HM 1/16W (0402) 5%TA-I/EM04...

SaaS 的迷茫



- 桌面程序移到 web 不是想象的那么简单
 - UED 的痛苦（浏览器兼容）
 - 后台架构的痛苦
 - 非常多的 js 代码导致 IE 执行效率低下
 - 系统运维的悲剧
- 承担网络问题而怪罪到软件
- 客户对商业模式的理解
 - 新鲜事物，都在观望
 - 免费的接受，收费的不接受
 - 行业软件的盈利模式（广告的可行性？）



云时代的 Python



- 底层架构里面遍地是 Python
- PaaS 平台出现了 GAE、SAE、ACE 等
- 各种 web 框架盛行：Django、Tornado...
- 创业公司（云用户）纷纷追随 Python

@逆雪寒 ：刚听到奇虎360的人力资源总监郑妈妈说，目前python的无论是再 LINUX 还是win上开发。事40万年就一个问题?????????????????
我靠~~ 把我吓醒了...

那你还等什么呢

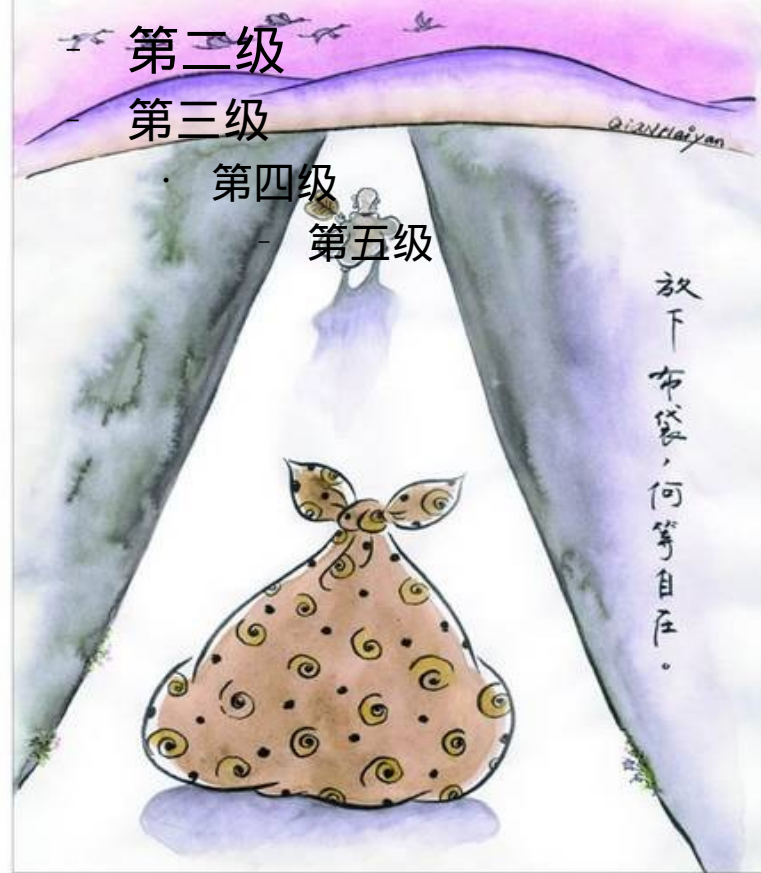
10月27日 22:47

来自新浪微博

转发(136) | 评论(43)



· 单击此处编辑母版文本样式



Thank you!



@ 网络化制造



Leejd80@gmail.c
om