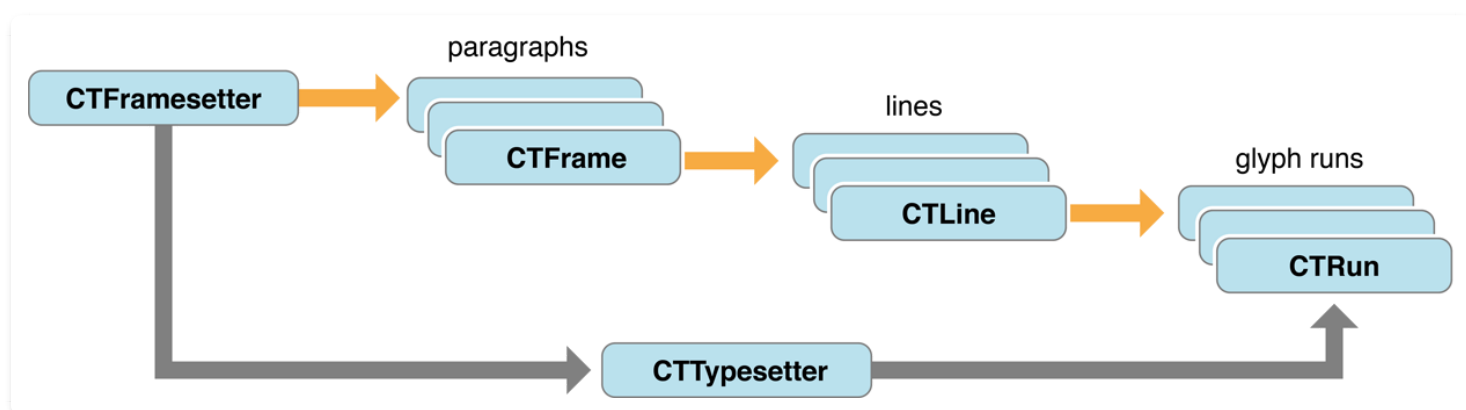


CoreText

简介

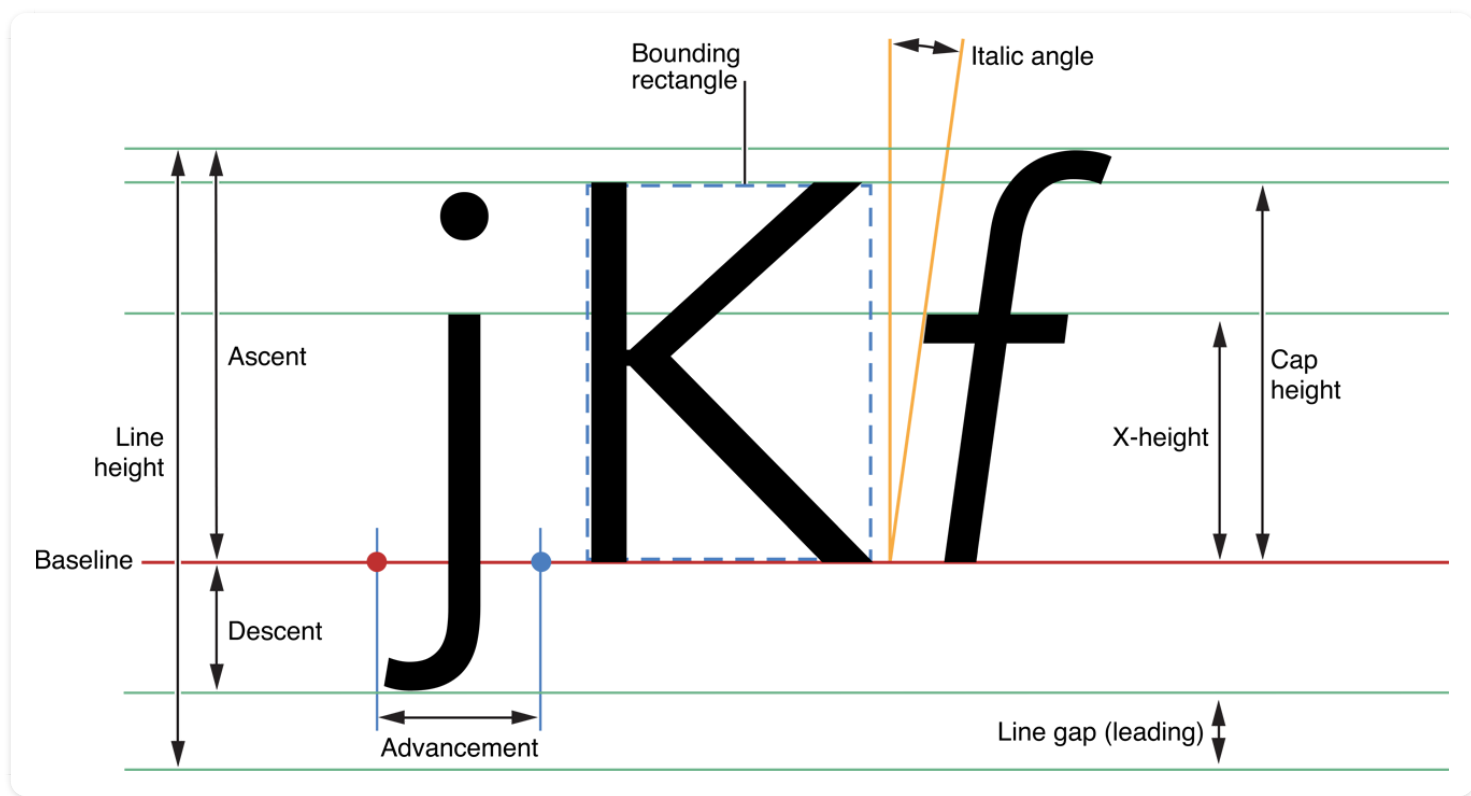
Core Text是用来layout文字与处理字体的先进与底层的技术。Core Text与Core Graphics(又称为Quartz)直接工作。Quartz能够直接处理字体(font)和字形(glyphs),将文字渲染到界面上,它是基础库中能够唯一处理字形的模块。因此,CoreText为了排版,需要将显示的文本内容、位置、字体、字形直接传递给Quartz。相比其他UI组件,而且由于Core Text提供了Quartz可直接使用的所有数据,因此结果用它显示就是高性能的文本渲染。Core Text functions 函数可以多个线程同时调用,只用调用者不改变任何参数。Core Text API是C语言的。

CoreText layout engine 架构



CTFramesetter是用于创建CTFrame的工厂,给CTFramesetter一个CGPath,它就会通过它持有的CTTypesetter生成CTFrame,一个CTFrame是由一行一行的CTLine的组成,每个CTLine又包含若干个CTRun(既字形绘制的最小单元),通过相应的方法可以获取到不同位置的CTRun和CTLine。平时写的NSAttributedString 通过CoreText提供的方法生成CTFramesetter,这里的CTTypesetter,CTFramesetter,CTRun,CTLine,CTFrame,与其他关联的layout对象都不是线程安全的。只能在一个操作、队列、线程中使用。

iOS 字形



Bounding rectangle: 虚拟矩形边框，容纳整个字体；
BaseLine: 基准线，文字以此为基础进行渲染；
Ascent: 文字的最高点距离基准线的距离，为正数；
Decent: 文字的最低点距离基准线的距离，为负数；
Leading: 行与行之间的距离；
一个文字行高 $\text{lineHeight} = \text{Ascent} + |\text{Decent}| + \text{Leading}$ 。

图文混排

CoreText没有相应API直接将一个图片转换为CTRun并进行绘制，它所能做的只是为图片预留响应的空白区域，而真正的绘制则是交由CoreGraphics完成。

在CoreText中提供了CTRunDelegate这么个CoreFoundation类，顾名思义它可以对CTRun进行拓展。AttributedString某个段设置kCTRunDelegateAttributeName属性之后，CoreText使用它生成CTRun是通过当前Delegate的回调来获取自己的ascent, descent和width，而不是根据文字信息。这样就给我们留下了可操作的空间：用一个空白字符作为图片的占位符，设好Delegate，占好位置，然后用CoreGraphics进行图片的绘制。

CTRunDelegate

图文混排预留的位置可以使用CTRunDelegate，CTRunDelegate控制排版特点像glyph ascent, glyph descent, 与glyph width。通过设置宽度，上下两个基准距离，为当前排版对象留出位置,当然这个

对象可以是UIImage或者UIView或者CALayer。UIImage将由CGContext画出, UIView或者CALayer将被作为text container的 view 或者layer。

```
//create CTRunDelegate
CTRunDelegateCallbacks callback;
    callback.version = kCTRunDelegateCurrentVersion;
    callback.dealloc = deallocCallback;
    callback.getWidth = getWidthCallback;
    callback.getAscent = getAscentCallback;
    callback.getDescent = getDecentCallback;
    NSDictionary *ref = @{@"width":@100,
                           @"ascent":@20,
                           @"decent":@10,};
    CTRunDelegateRef runDelegate = CTRunDelegateCreate(&callback, (__bridge_retained void *)ref);

//设置好需要的NSAttributedString绑定
[attrStr addAttribute:(id)kCTRunDelegateAttributeName
                  value:(__bridge id)runDelegate
                  range:NSMakeRange(0, 1)];
```

在渲染的时候将图片渲染到图层上面。

NSParagraphStyle

NSParagraphStyle与attributed string一起使用设置段落，NSParagraphStyle中的所有属性只是可读， 要设置他们可以用NSMutableParagraphStyle子类。

```
lineSpacing 行距
paragraphSpacing 段间距(与下一段落)
alignment 对齐方式
firstLineHeadIndent 首行缩进
headIndent 头部缩进
tailIndent 尾部缩进
lineBreakMode 结尾以.....方式省略
minimumLineHeight 最低行高
maximumLineHeight 最高行高
baseWritingDirection 书写方向 (3种, 自定义, 从左到右, 从右到左)
lineHeightMultiple 行间距是多少倍(要正数);
paragraphSpacingBefore 段首行间距 (与上一段落)
hyphenationFactor 连字属性 有效值介于0.0与1.0
```