实验编号： 13 **四川师大《IOS》实验报告 2018** 年 **12** 月 **12** 日

### **计算机科学学院** 2016 级 4 班 实验名称： 多线程和网络程序设计 \_

姓名：蒋星 学号：2016110418 指导老师：\_\_李贵洋\_\_ 实验成绩:\_\_\_\_\_

**实验 十三 \_**多线程和网络程序设计**\_\_\_\_\_**

1. 实验目的及要求
2. 理解并掌握iOS多线程编程的相关技术；
3. 掌握GCD关键技术，包括block、dispatch等；
4. 掌握WebView的使用；
5. 掌握URLSession的使用，
6. 掌握第三网络库Alamofire的使用方法；
7. 掌握Json的解析。
8. 实验要求
9. 认真填写实验报告，要求附加部分运行界面和主要代码；
10. 对设计好的程序，检查输出是否符合预期，如有错请分析错误原因并解决；
11. 实验内容
12. 采用多线程技术，实现一个大数加程序。
    1. 正确理解DispatchQueue的使用
    2. 从1 到 9999999
    3. 不能阻塞UI主线程
13. Web浏览器;
    1. 使用WebView控件写成一个简易的浏览器，有浏览器的基本功能；
14. 使用网络库进行天气Json数据的解析
    1. APP有两个界面，第一个界面：tableview显示一个城市列表
    2. 第二个界面，显示选择城市的天气数据
    3. 使用第三方网络Alamofire进行网络的连接，获取网络天气数据；
    4. 对获取到的网络数据进行Json的解析；
    5. 天气数据库位置：http://t.weather.sojson.com/api/weather/city/101270101

可选其他 Web API 进行解析:

1. 免费 JSON API: <http://www.sojson.com/api/>
2. 源代码管理 GitHub API: <https://developer.github.com/v3/>
3. 图形识别: <https://imagga.com/>

4. 摄影社区: <https://500px.com/>

1. 实验主要流程、基本操作或核心代码、算法片段（该部分如不够填写，请另加附页）
2. 采用多线程技术，实现一个大数加程序。
   1. 正确理解DispatchQueue的使用
   2. 从1 到 9999999
   3. 不能阻塞UI主线程

* 程序代码：

@IBOutlet weak var lbltime: UILabel!

@IBOutlet weak var result: UILabel!

@IBOutlet weak var counter: UILabel!

var count = 0

override func viewDidLoad() {

super.viewDidLoad()

Timer.scheduledTimer(withTimeInterval: 1, repeats: true) { (timer) in

self.count += 1

self.counter.text = "计时器：\(self.count)"

print("timer thread: \(Thread.current)")

}

// Do any additional setup after loading the view, typically from a nib.

}

override func didReceiveMemoryWarning() {

super.didReceiveMemoryWarning()

// Dispose of any resources that can be recreated.

}

@IBAction func getCalRsult(\_ sender: UIButton) {

var sum = 0

DispatchQueue.global().async {

print("sum thread: \(Thread.current)")

for i in 1...9999999{

sum += i

}

//ui 操作只能放倒主线程中

DispatchQueue.main.async {

self.result.text = "\(sum)"

}

}

}

* 运行结果：



1. Web浏览器;
   1. 使用WebView控件写成一个简易的浏览器，有浏览器的基本功能；

* 程序代码：

override func viewDidLoad() {

super.viewDidLoad()

if let url = URL(string:"https://www.hao123.com/"){

uiweb.load(URLRequest(url:url))

}

// Do any additional setup after loading the view, typically from a nib.

}

@IBAction func back(\_ sender: UIButton) {

uiweb.goBack()

}

@IBAction func forward(\_ sender: UIButton) {

uiweb.goForward()

}

@IBAction func reload(\_ sender: UIButton) {

uiweb.reload()

}

@IBAction func gotoload(\_ sender: Any) {

if let url = URL(string:tfurl.text!){

uiweb.load(URLRequest(url:url))

}

}

* 运行结果：



1. 使用网络库进行天气Json数据的解析
   1. APP有两个界面，第一个界面：tableview显示一个城市列表
   2. 第二个界面，显示选择城市的天气数据
   3. 使用第三方网络Alamofire进行网络的连接，获取网络天气数据；
   4. 对获取到的网络数据进行Json的解析；
   5. 天气数据库位置：http://t.weather.sojson.com/api/weather/city/101270101

* 程序代码：

//北京: 101010100

//成都: 101270101

//重庆: 101040100

//上海: 101020100

//天津: 101030100

//黑龙江:101050101

let citycode:[String:String] = ["北京":"101010100","成都":"101270101","重庆":"101040100","上海":"101020100","天津":"101030100","黑龙江":"101050101"]

var cityinfo:Cityinfo?

var sendcityname:String?

@IBOutlet weak var cityname: UILabel!

@IBOutlet weak var pm25status: UILabel!

@IBOutlet weak var shidustatus: UILabel!

@IBOutlet weak var wendu: UILabel!

@IBOutlet weak var updatetime: UILabel!

@IBOutlet weak var nowtime: UILabel!

@IBOutlet weak var quality: UILabel!

@IBOutlet weak var advice: UILabel!

override func viewDidLoad() {

super.viewDidLoad()

let url\_ = "http://t.weather.sojson.com/api/weather/city/\(citycode[sendcityname!] ?? "101270101")"

let url = URL(string:url\_)!

AF.request(url).responseJSON {(response) in

self.cityinfo = Cityinfo()

// print(response.data!) //没有序列化的json 数据

// print(response.result.value!) // 序列化的json

let json\_ = response.result.value as! Dictionary<String,AnyObject>

let cityInfo = json\_["cityInfo"] as! Dictionary<String,AnyObject>

let todayinfo = json\_["data"] as! Dictionary<String,AnyObject>

// 2.得到湿度

self.cityinfo?.shidu = todayinfo["shidu"] as! String

// 3.得到pm25

self.cityinfo?.pm25 = "\(todayinfo["pm25"] as! Int)"

// 4.得到建议

self.cityinfo?.ganmaoadvice = todayinfo["ganmao"] as! String

// 5.得到时间

self.cityinfo?.time = json\_["time"] as! String

// 6.得到空气质量

self.cityinfo?.quality = todayinfo["quality"] as! String

// 7.得到更新时间

self.cityinfo?.updateTime = cityInfo["updateTime"] as! String

// 8.获取温度

self.cityinfo?.wendu = todayinfo["wendu"] as! String

self.update()

}

// Do any additional setup after loading the view.

}

func update(){

cityname.text = "城市:" + sendcityname!

pm25status.text = "PM2.5:" + (cityinfo?.pm25)!

shidustatus.text = "湿度:" + (cityinfo?.shidu)!

wendu.text = "温度:" + (cityinfo?.wendu)!

updatetime.text = "更新时间:" + (cityinfo?.updateTime)!

nowtime.text = "当前时间:" + (cityinfo?.time)!

quality.text = "空气质量:" + (cityinfo?.quality)!

advice.text = "气象建议:" + (cityinfo?.ganmaoadvice)!

}

* 运行结果：





可选其他 Web API 进行解析:

1. 免费 JSON API: <http://www.sojson.com/api/>
2. 源代码管理 GitHub API: <https://developer.github.com/v3/>
3. 图形识别: <https://imagga.com/>

4. 摄影社区: <https://500px.com/>

1. 实验结果的分析与评价（该部分如不够填写，请另加附页）

Github地址：https://github.com/KikcerGoer/study-swift/tree/master/homework

真是难做人

注：实验成绩等级分为（90－100分）优，（80－89分）良，(70-79分)中，（60－69分）及格，（59分）不及格。