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： 郭周倩 学号： 2016110413 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. 认真填写实验报告，要求附加部分运行界面和主要代码；
9. 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；
10. 实验内容
11. 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
12. 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；
13. 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：

import UIKit

class ViewController: UIViewController {

    @IBOutlet weak var sumlabel: UILabel!

    @IBOutlet weak var timerlabel: UILabel!

    var count = 0

    override func viewDidLoad() {

        super.viewDidLoad()

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

            self.count += 1

            self.timerlabel.text = "time:\(self.count)"

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

        }

    }

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

        var sum = 0

        DispatchQueue.global().async {

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

            for i in 1...999999999{

                sum += i

            }

            DispatchQueue.main.async {

                self.sumlabel.text = "sum:\(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：

import UIKit

import WebKit

class ViewController: UIViewController {

    @IBOutlet weak var text: UITextField!

    @IBOutlet weak var webView: WKWebView!

    override func viewDidLoad() {

        super.viewDidLoad()

        if let url = URL(string: "http://www.163.com") {

            print("aaaaa")

            webView.load(URLRequest(url: url))

            //            webView.addObserver(self, forKeyPath: "estimatedProgress", options: .new, context: nil)

        }

    }

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

        if let url = URL(string: text.text ?? "") {

            webView.load(URLRequest(url: url))

        }

    }

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

        webView.goBack()

    }

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

        webView.goForward()

    }

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

        webView.reload()

    }

}

* 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：

import UIKit

import Alamofire

class JSONViewController: UIViewController {

    let url = URL(string: "http://t.weather.sojson.com/api/weather/city/101270101")!

    var citys:[String:Any]?

    override func viewDidLoad() {

        super.viewDidLoad()

        // Do any additional setup after loading the view.

    }

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

        if let data = try? Data(contentsOf: url){

            if let json = try? JSONSerialization.jsonObject(with: data, options: .allowFragments) as! [String:Any]{

                let str:[String:String] = json["cityInfo"] as! [String:String]

                print(str["city"] ?? "")

                print(json["data"] ?? "")

            }

        }

    }

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

        let task = URLSession.shared.dataTask(with: url) { (data, response, error) in

            if let json = try? JSONSerialization.jsonObject(with: data!, options: .allowFragments) as! [String:Any]{

                self.citys = json

                /\*let str:[String:String] = json["cityInfo"] as! [String:String]

                print(str["city"] ?? "")

                print(json["data"] ?? "")\*/

                DispatchQueue.main.async {

                    self.performSegue(withIdentifier: "ShowCityList", sender: self)

                }

            }

        }

        task.resume()

    }

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

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

            self.citys = response.value as? [String:Any]

           /\* let str:[String:String] = json["cityInfo"] as! [String:String]

            print(str["city"] ?? "")

            print(json["data"] ?? "")\*/

            self.performSegue(withIdentifier: "ShowCityList", sender: self)

        }

    }

    // MARK: - Navigation

    // In a storyboard-based application, you will often want to do a little preparation before navigation

    override func prepare(for segue: UIStoryboardSegue, sender: Any?) {

        // Get the new view controller using segue.destination.

        // Pass the selected object to the new view controller.

        if segue.identifier == "ShowCityList" {

            if let secVC = segue.destination as? PersonsTableViewController {

                secVC.citys = self.citys

            }

        }

}

}

import UIKit

class PersonsTableViewController: UITableViewController {

    var citys:[String:Any]?

    override func viewDidLoad() {

        super.viewDidLoad()

        // Uncomment the following line to preserve selection between presentations

        // self.clearsSelectionOnViewWillAppear = false

        // Uncomment the following line to display an Edit button in the navigation bar for this view controller.

        // self.navigationItem.rightBarButtonItem = self.editButtonItem

    }

    // MARK: - Table view data source

    override func numberOfSections(in tableView: UITableView) -> Int {

        // #warning Incomplete implementation, return the number of sections

        return 1

    }

    override func tableView(\_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {

        // #warning Incomplete implementation, return the number of rows

//        return citys?.count ?? 0

        return 1   //json中只有1个城市

    }

    override func tableView(\_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {

        let cell = tableView.dequeueReusableCell(withIdentifier: "Cell", for: indexPath)

        // Configure the cell...

        let str:[String:String] = citys?["cityInfo"] as! [String:String]

        cell.textLabel?.text = "\(str["parent"] ?? "")"

        cell.detailTextLabel?.text = "\(str["city"] ?? "")"

        return cell

    }

    /\*

    // Override to support conditional editing of the table view.

    override func tableView(\_ tableView: UITableView, canEditRowAt indexPath: IndexPath) -> Bool {

        // Return false if you do not want the specified item to be editable.

        return true

    }

    \*/

    /\*

    // Override to support editing the table view.

    override func tableView(\_ tableView: UITableView, commit editingStyle: UITableViewCellEditingStyle, forRowAt indexPath: IndexPath) {

        if editingStyle == .delete {

            // Delete the row from the data source

            tableView.deleteRows(at: [indexPath], with: .fade)

        } else if editingStyle == .insert {

            // Create a new instance of the appropriate class, insert it into the array, and add a new row to the table view

        }

    }

    \*/

    /\*

    // Override to support rearranging the table view.

    override func tableView(\_ tableView: UITableView, moveRowAt fromIndexPath: IndexPath, to: IndexPath) {

    }

    \*/

    /\*

    // Override to support conditional rearranging of the table view.

    override func tableView(\_ tableView: UITableView, canMoveRowAt indexPath: IndexPath) -> Bool {

        // Return false if you do not want the item to be re-orderable.

        return true

    }

    \*/

    // MARK: - Navigation

    // In a storyboard-based application, you will often want to do a little preparation before navigation

    override func prepare(for segue: UIStoryboardSegue, sender: Any?) {

        // Get the new view controller using segue.destination.

        // Pass the selected object to the new view controller.

        if  let secVC = segue.destination as? PersonViewController {

            secVC.city = self.citys

        }

    }

}

import UIKit

class PersonViewController: UIViewController {

    @IBOutlet weak var shidu: UILabel!

    @IBOutlet weak var pm25: UILabel!

    @IBOutlet weak var pm10: UILabel!

    @IBOutlet weak var quality: UILabel!

    @IBOutlet weak var wendu: UILabel!

    @IBOutlet weak var ganmao: UILabel!

    var city:[String:Any]?

    override func viewDidLoad() {

        super.viewDidLoad()

        let str:[String:Any] = city?["data"] as! [String:Any]

        shidu.text = str["shidu"] as? String

        pm25.text = "\(str["pm25"] ?? "")"

        pm10.text = "\(str["pm10"] ?? "")"

        quality.text = str["quality"] as? String

        wendu.text = str["wendu"] as? String

        ganmao.text = str["ganmao"] as? String

        print(str["pm25"] ?? "")

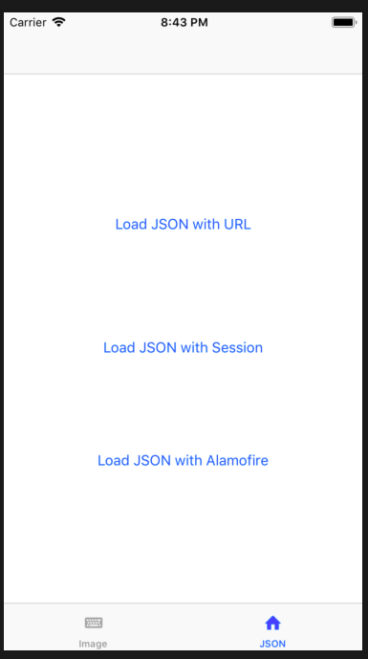
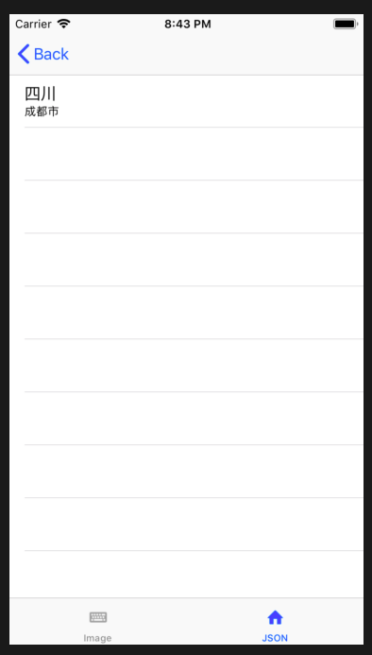
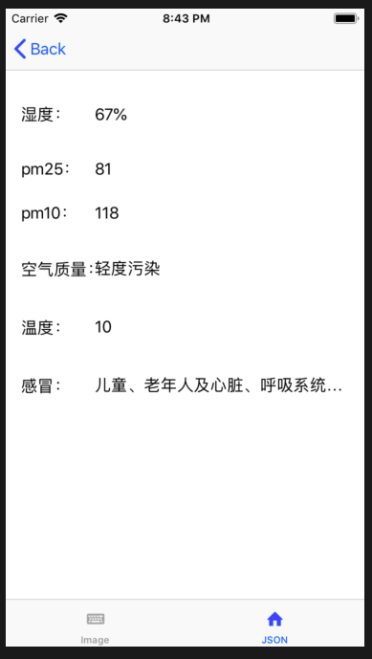
        print(str["pm10"] ?? "")

        // Do any additional setup after loading the view.

    }

}

* 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/Gzqqqqq/IOSExperimentReport.git>

本次实验的核心放在多线程的处理，以及json数据的传值这两个方面。多线程的处理是所学的多线程中最简单的一种，只需要一个回掉函数调出或者调入主线程中。而网上获取json值有三种方法，最简单的一种就是Alamofire，自动转换json数据，我们需要做的就是获取到的json数据传递给其他界面。

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。