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：\_樊琳\_ 学号：\_2016110408\_ 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. 实验内容
9. 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
10. 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；
11. 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：

//

// AppDelegate.swift

// 13.1

//

// Created by student on 2018/12/12.

// Copyright © 2018年 fl. All rights reserved.

//

import UIKit

@UIApplicationMain

class AppDelegate: UIResponder, UIApplicationDelegate {

var window: UIWindow?

func application(\_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {

// Override point for customization after application launch.

return true

}

func applicationWillResignActive(\_ application: UIApplication) {

// Sent when the application is about to move from active to inactive state. This can occur for certain types of temporary interruptions (such as an incoming phone call or SMS message) or when the user quits the application and it begins the transition to the background state.

// Use this method to pause ongoing tasks, disable timers, and invalidate graphics rendering callbacks. Games should use this method to pause the game.

}

func applicationDidEnterBackground(\_ application: UIApplication) {

// Use this method to release shared resources, save user data, invalidate timers, and store enough application state information to restore your application to its current state in case it is terminated later.

// If your application supports background execution, this method is called instead of applicationWillTerminate: when the user quits.

}

func applicationWillEnterForeground(\_ application: UIApplication) {

// Called as part of the transition from the background to the active state; here you can undo many of the changes made on entering the background.

}

func applicationDidBecomeActive(\_ application: UIApplication) {

// Restart any tasks that were paused (or not yet started) while the application was inactive. If the application was previously in the background, optionally refresh the user interface.

}

func applicationWillTerminate(\_ application: UIApplication) {

// Called when the application is about to terminate. Save data if appropriate. See also applicationDidEnterBackground:.

}

}

//

// ViewController.swift

// 13.1

//

// Created by student on 2018/12/12.

// Copyright © 2018年 fl. All rights reserved.

//

import UIKit

class ViewController: UIViewController {

@IBOutlet weak var timerLabel: UILabel!

@IBOutlet weak var testLabel: UILabel!

var count = 0

override func viewDidLoad() {

super.viewDidLoad()

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

self.count += 1

self.timerLabel.text = "\(self.count)"

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

}.fire()

}

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

var sum = 0

for \_ in 1...100 {

DispatchQueue.global().async {

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

for i in 1...1000 {

usleep(10000)

sum += i

}

DispatchQueue.main.async {

self.testLabel.text = "sum:\(sum)"

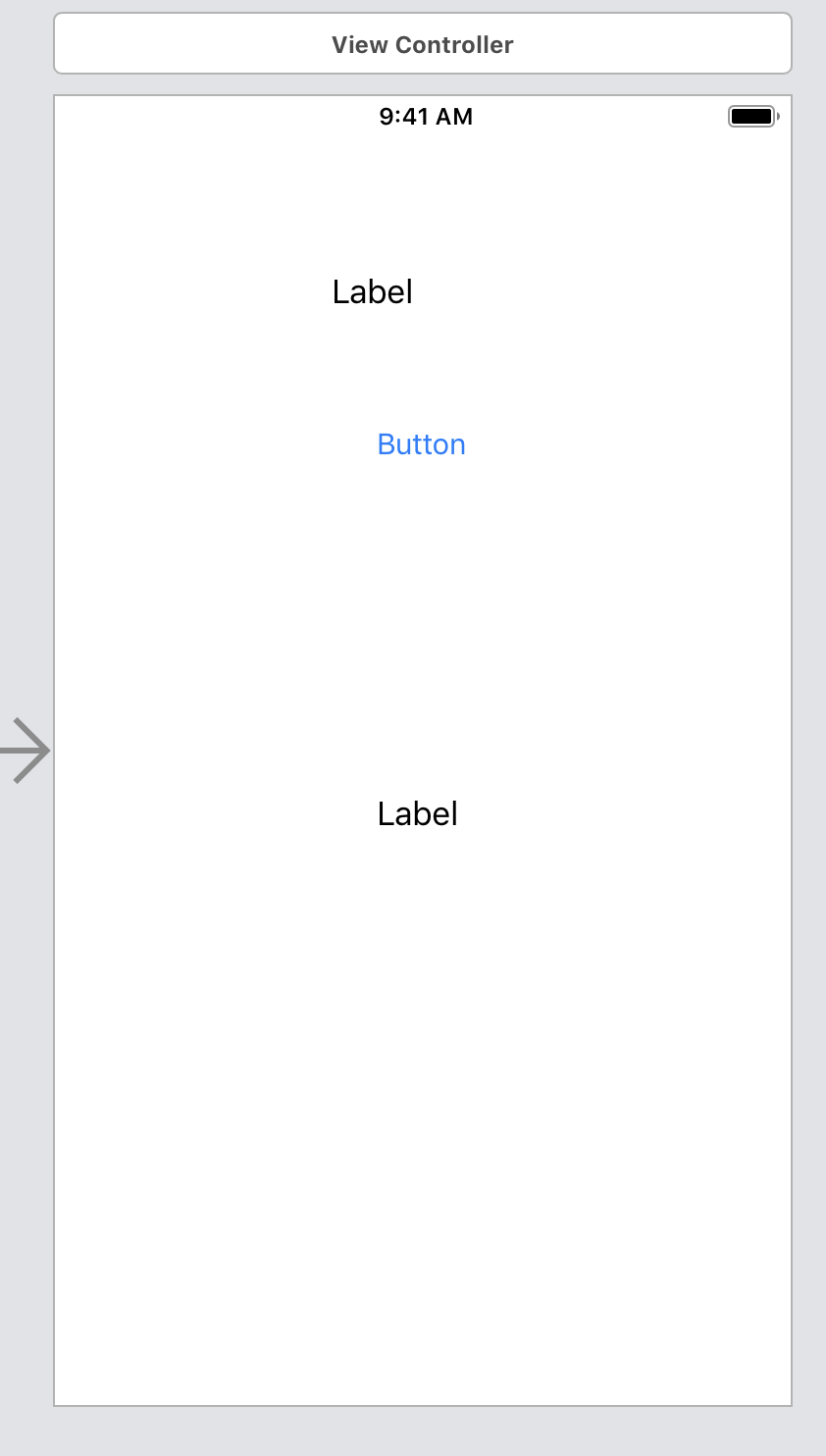
}

}

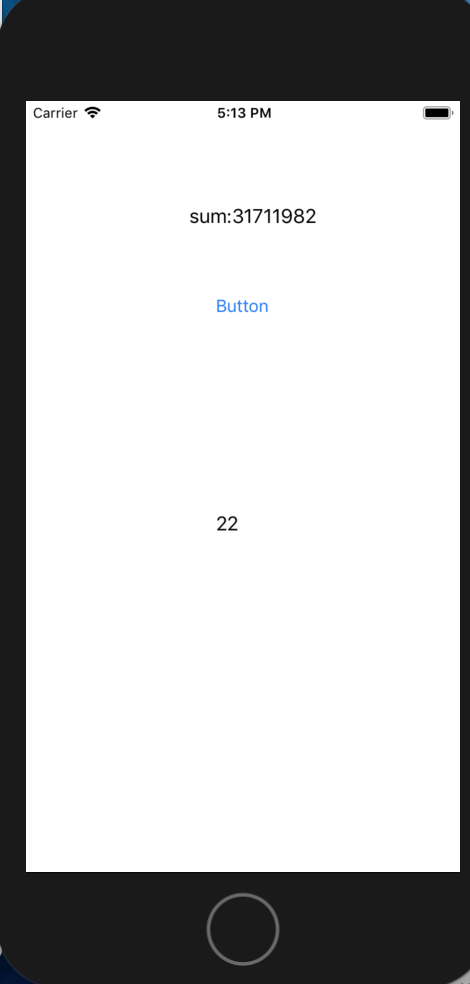
}

}

}



* 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：

//

// AppDelegate.swift

// 13.2

//

// Created by student on 2018/12/12.

// Copyright © 2018年 fl. All rights reserved.

//

import UIKit

@UIApplicationMain

class AppDelegate: UIResponder, UIApplicationDelegate {

var window: UIWindow?

func application(\_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {

// Override point for customization after application launch.

return true

}

func applicationWillResignActive(\_ application: UIApplication) {

// Sent when the application is about to move from active to inactive state. This can occur for certain types of temporary interruptions (such as an incoming phone call or SMS message) or when the user quits the application and it begins the transition to the background state.

// Use this method to pause ongoing tasks, disable timers, and invalidate graphics rendering callbacks. Games should use this method to pause the game.

}

func applicationDidEnterBackground(\_ application: UIApplication) {

// Use this method to release shared resources, save user data, invalidate timers, and store enough application state information to restore your application to its current state in case it is terminated later.

// If your application supports background execution, this method is called instead of applicationWillTerminate: when the user quits.

}

func applicationWillEnterForeground(\_ application: UIApplication) {

// Called as part of the transition from the background to the active state; here you can undo many of the changes made on entering the background.

}

func applicationDidBecomeActive(\_ application: UIApplication) {

// Restart any tasks that were paused (or not yet started) while the application was inactive. If the application was previously in the background, optionally refresh the user interface.

}

func applicationWillTerminate(\_ application: UIApplication) {

// Called when the application is about to terminate. Save data if appropriate. See also applicationDidEnterBackground:.

}

}

//

// ViewController.swift

// 13.1

//

// Created by student on 2018/12/12.

// Copyright © 2018年 fl. All rights reserved.

//

import UIKit

import WebKit

class ViewController: UIViewController,WKNavigationDelegate,WKUIDelegate {

@IBOutlet weak var tfUrl: UITextField!

@IBOutlet weak var webView: WKWebView!

@IBOutlet weak var progressView: UIProgressView!

override func viewDidLoad() {

super.viewDidLoad()

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

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

webView.load(URLRequest(url: url))

webView.navigationDelegate = self

webView.uiDelegate = self

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

}

}

func webView(\_ webView: WKWebView, didFinish navigation: WKNavigation!) {

webView.evaluateJavaScript("document.title") { (title, error) in

if let title = title as? String {

self.title = title

}

}

webView.evaluateJavaScript("ocToJs('test','i hate you')", completionHandler: {(response, error) in

print("response:\((response as? String) ?? "") error:\(error?.localizedDescription ?? "")")

})

webView.evaluateJavaScript("showAlert()", completionHandler: {(response, error) in

print("response:\((response as? String) ?? "") error:\(error?.localizedDescription ?? "")")

})

print("Finished!")

}

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

if let url = URL(string: tfUrl.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()

}

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

if let url = Bundle.main.url(forResource: "default", withExtension: "html") {

// webView.load(URLRequest(url: url))

webView.loadFileURL(url, allowingReadAccessTo: url)

}

}

func webView(\_ webView: WKWebView, runJavaScriptAlertPanelWithMessage message: String, initiatedByFrame frame: WKFrameInfo, completionHandler: @escaping () -> Void) {

let alert = UIAlertController(title: "Alert", message: message, preferredStyle: UIAlertController.Style.alert)

alert.addAction(UIAlertAction(title: "Cancel", style: .cancel, handler: { action in

completionHandler()

}))

present(alert, animated: true, completion: nil)

}

func webView(\_ webView: WKWebView, runJavaScriptConfirmPanelWithMessage message: String, initiatedByFrame frame: WKFrameInfo, completionHandler: @escaping (Bool) -> Void) {

let alert = UIAlertController(title: "Alert", message: message, preferredStyle: UIAlertController.Style.alert)

alert.addAction(UIAlertAction(title: "Cancel", style: .cancel, handler: { (action) in

completionHandler(false)

}))

alert.addAction(UIAlertAction(title: "Ok", style: .default, handler: { (action) in

completionHandler(true)

}))

present(alert, animated: true, completion: nil)

}

func webView(\_ webView: WKWebView, runJavaScriptTextInputPanelWithPrompt prompt: String, defaultText: String?, initiatedByFrame frame: WKFrameInfo, completionHandler: @escaping (String?) -> Void) {

let alert = UIAlertController(title: prompt, message: nil, preferredStyle: UIAlertController.Style.alert)

alert.addTextField { (textField) in

textField.text = defaultText

}

alert.addAction(UIAlertAction(title: "Ok", style: .default, handler: { (action) in

completionHandler(alert.textFields?.first?.text ?? "nil")

}))

present(alert, animated: true, completion: nil)

}

override func observeValue(forKeyPath keyPath: String?, of object: Any?, change: [NSKeyValueChangeKey : Any]?, context: UnsafeMutableRawPointer?) {

if keyPath == "estimatedProgress" {

progressView.alpha = 1

progressView.progress = Float(webView.estimatedProgress)

if progressView.progress == 1 {

progressView.progress = 0

progressView.alpha = 0

}

}

}

deinit {

webView.removeObserver(self, forKeyPath: "estimatedProgress")

}

}

* 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：

//

// AppDelegate.swift

// weather

//

// Created by student on 2018/12/12.

// Copyright © 2018年 fl. All rights reserved.

//

import UIKit

@UIApplicationMain

class AppDelegate: UIResponder, UIApplicationDelegate {

var window: UIWindow?

func application(\_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {

// Override point for customization after application launch.

window?.rootViewController = UINavigationController(rootViewController: ViewController())

return true

}

func applicationWillResignActive(\_ application: UIApplication) {

// Sent when the application is about to move from active to inactive state. This can occur for certain types of temporary interruptions (such as an incoming phone call or SMS message) or when the user quits the application and it begins the transition to the background state.

// Use this method to pause ongoing tasks, disable timers, and invalidate graphics rendering callbacks. Games should use this method to pause the game.

}

func applicationDidEnterBackground(\_ application: UIApplication) {

// Use this method to release shared resources, save user data, invalidate timers, and store enough application state information to restore your application to its current state in case it is terminated later.

// If your application supports background execution, this method is called instead of applicationWillTerminate: when the user quits.

}

func applicationWillEnterForeground(\_ application: UIApplication) {

// Called as part of the transition from the background to the active state; here you can undo many of the changes made on entering the background.

}

func applicationDidBecomeActive(\_ application: UIApplication) {

// Restart any tasks that were paused (or not yet started) while the application was inactive. If the application was previously in the background, optionally refresh the user interface.

}

func applicationWillTerminate(\_ application: UIApplication) {

// Called when the application is about to terminate. Save data if appropriate. See also applicationDidEnterBackground:.

}

}

import UIKit

import Alamofire

import SwiftyJSON

class ViewController: UIViewController, UITableViewDelegate, UITableViewDataSource {

var tableView: UITableView!

let citise = ["成都": 101270101]

override func viewDidLoad() {

super.viewDidLoad()

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

self.view.backgroundColor = UIColor.white

self.title = "cities"

tableView = UITableView(frame: self.view.frame)

tableView.delegate = self

tableView.dataSource = self

self.view.addSubview(tableView)

}

func tableView(\_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {

let site = "http://t.weather.sojson.com/api/weather/city/101270101"

let url = URL(string: site)!

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

if let json = try? JSON(data: response.result.value as! Data) {

let dict = json["weatherinfo"]

let city = dict["city"].string!

let temp = dict["temp"].string!

let wd = dict["WD"].string!

let ws = dict["WS"].string!

var weather = "温度: \(temp)\n"

weather += "风向: \(wd)\n"

weather += "风力: \(ws)"

let viewController = WeaTherViewController()

viewController.navTitle = city

viewController.weather = weather

self.navigationController?.pushViewController(viewController, animated: true)

}

}

}

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

return citise.count

}

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

var cell = tableView.dequeueReusableCell(withIdentifier: "Cell")

if cell == nil {

cell = UITableViewCell(style: .default, reuseIdentifier: "Cell")

cell?.accessoryType = .disclosureIndicator

}

cell?.textLabel?.text = Array(citise.keys)[indexPath.row]

return cell!

}

override func didReceiveMemoryWarning() {

super.didReceiveMemoryWarning()

// Dispose of any resources that can be recreated.

}

}

//

// WeaTherViewController.swift

// weather

//

// Created by student on 2018/12/12.

// Copyright © 2018年 fl. All rights reserved.

//

import UIKit

class WeaTherViewController: UIViewController {

var weather: String?

var navTitle: String?

override func viewDidLoad() {

super.viewDidLoad()

self.view.backgroundColor = UIColor.white

self.title = navTitle

let label = UILabel(frame: CGRect(x: 10, y: 100, width: self.view.frame.width - 20, height: self.view.frame.height - 150))

label.text = weather!

label.textAlignment = .center

label.layer.borderWidth = 1

label.textColor = UIColor.black

label.numberOfLines = 0

label.font = UIFont.systemFont(ofSize: 36)

self.view.addSubview(label)

}

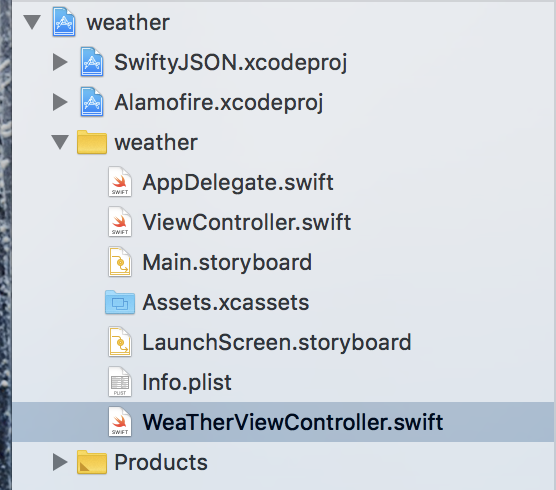
override func didReceiveMemoryWarning() {

super.didReceiveMemoryWarning()

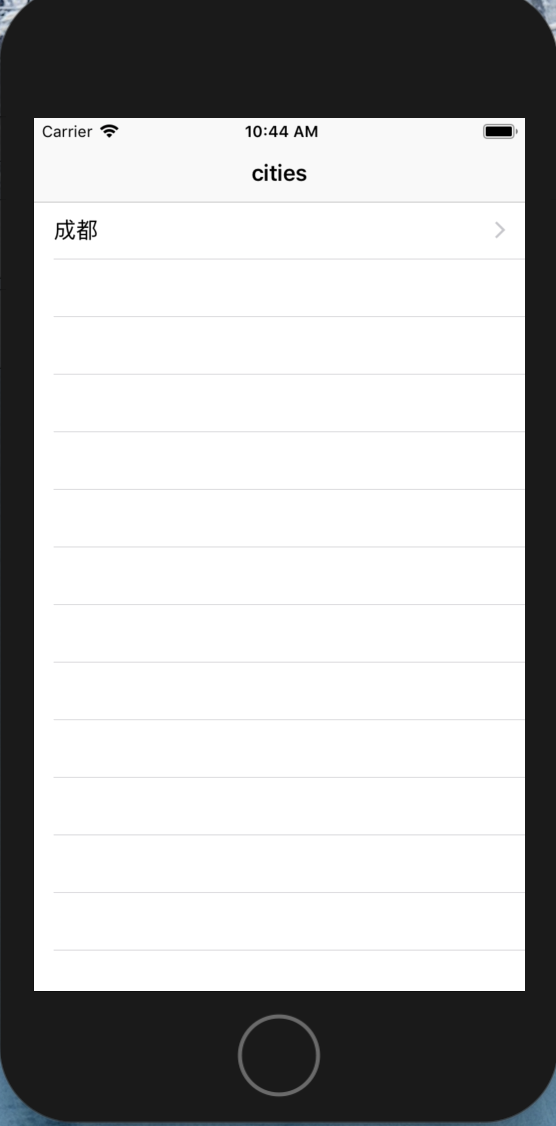
// Dispose of any resources that can be recreated.

}

}



* 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）

Swift语言多线程不难，分清楚main队列和其他自定义队列就可以。网络也还好，主要是json的运用对于我来说不是很熟练，要多加练习关于json这一块。

Github地址：

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。