

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG
-----o0o-----



ISO 9001 : 2008

NGHIÊN CỨU XÂY DỰNG ỨNG DỤNG CHO HỆ ĐIỀU HÀNH iOS

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
NGÀNH CÔNG NGHỆ THÔNG TIN**

Giáo viên hướng dẫn: Ths. Trần Ngọc Thái

Sinh viên : Hoàng Kim Ngọc

Lớp : CTL401

Hải Phòng, 7/2012

LỜI CẢM ƠN

Vậy là gần 5 năm đã trôi qua, mái trường Đại học Dân lập Hải Phòng thân thương cho em thật nhiều kỷ niệm sâu sắc mà em sẽ không thể nào quên.

Ngày ngày đến lớp, chúng em không chỉ được sống trong một môi trường học tập chuyên nghiệp, thu được những kiến thức bổ ích làm hành trang trên đường đời sau này mà còn được sống trong tình cảm quan tâm, triu mến của các thầy, cô.

Đối với những sinh viên năm cuối như chúng em, được làm đề án tốt nghiệp là một niềm vui, niềm hạnh phúc vô cùng lớn lao và đầy tự hào.

Để bài đề án được hoàn thành và có kết quả tốt như ngày hôm nay em xin gửi lời tri ân và lời cảm ơn sâu sắc nhất tới:

Thầy hiệu trưởng Trần Hữu Nghị.

Ban giám hiệu nhà trường cùng các thầy cô giáo bộ môn Công nghệ thông tin đã tận tình chỉ bảo, dạy dỗ, truyền đạt kiến thức cho chúng em nên người.

Và em xin dành lời cảm ơn đặc biệt từ tận đáy lòng đến thầy giáo, Th.s Trần Ngọc Thái. Trong suốt thời gian qua thầy đã giúp đỡ em rất nhiều, nếu không nhận được sự hướng dẫn của thầy có lẽ đề án tốt nghiệp của em không được hoàn thành thuận lợi như ngày hôm nay.

Bên cạnh đó, em cũng vô cùng biết ơn gia đình đã động viên, ủng hộ em khi em lựa chọn mái trường Dân Lập Hải Phòng là ngôi nhà thứ hai của mình.

Do kiến thức bản thân còn nhiều hạn chế nên bài đề án của em vẫn còn nhiều sai sót, vì vậy em mong nhận được sự góp ý của các thầy, cô để bài đề án của em được hoàn thiện hơn.

Em xin chân thành cảm ơn !

Hải Phòng, 8 tháng 07 năm 2012

Sinh viên

Hoàng Kim Ngọc

MỤC LỤC

DANH MỤC CÁC HÌNH	4
MỞ ĐẦU	5
CHƯƠNG 1: KIẾN TRÚC HỆ ĐIỀU HÀNH iOS	6
1.1.Định nghĩa iOS :.....	6
1.2.Tổng quan về kiến trúc của iOS:	6
1.2.1 .Các tiến trình (Processes):	7
1.2.2.Nhân (Kernel):	7
1.2.3.Bộ đệm gói (Packet buffer):	7
1.2.4.Trình điều khiển thiết bị (Device driver):.....	7
1.2.5.Phần mềm chuyển mạch nhanh (Fast switching soft):	7
1.3.Tổ chức bộ nhớ:	7
1.3.1.Miền bộ nhớ (Memory region) :.....	7
1.3.1.Miền bộ nhớ (Memory region) :	8
1.3.2.Vùng bộ nhớ (Memory pool):	9
1.4.Tiến trình iOS (Processes iOS):	10
1.4.1.Vòng đời của một tiến trình:	10
1.4.1.1.Trạng thái khởi tạo (Create):	11
1.4.1.2.Trạng thái điều chỉnh (Modify):	11
1.4.1.3.Trạng thái thực thi (Execute):	11
1.4.1.4.Trạng thái kết thúc (Terminal):	12
1.4.2.Độ ưu tiên tiến trình iOS:	12
1.5.Kernel iOS:	13
1.5.1.Lập lịch:	13
1.5.2.Quản lý bộ nhớ:	13
1.5.2.1.Bộ quản lý Region:	14
1.5.2.2.Bộ quản lý pool:	14
1.5.2.2.Quản lý Chunk:	15
1.5.3.Quản lý bộ đệm gói:	15
1.6.Trình điều khiển thiết bị:	16

1.7.Kiến trúc của hệ điều hành iPhone :	18
1.7.1.Lớp Core OS :	18
1.7.2.Lớp Core Services :	19
1.7.3.Lớp Media(Truyền thông) :	19
1.7.4.Lớp Cocoa Touch :	20
CHƯƠNG 2: PHÁT TRIỂN ỨNG DỤNG THI HÀNH TRÊN iOS	21
2.1.Lập trình Xcode :	21
2.1.1.Xcode IDE (Môi trường phát triển tích hợp) :	22
2.1.1.1. <i>Single Window(Cửa sổ đơn)</i> :	23
2.1.1.2. <i>Navigators</i> :	24
2.1.1.3. <i>Jump Bar</i> :	24
2.1.2.Interface Builder(Giao diện chương trình):	24
2.1.3.Apple LLVM(Trình biên dịch) :	25
2.2. Iphone Simulator(Mô phỏng Iphone) :	25
2. 3.Instrument :	27
2.4. iOS Framework :	27
2.5.Một số ứng dụng đơn giản :	31
2.5.1.Chương trình Hello World :	31
2.5.2.Đóng gói chương trình:	35
CHƯƠNG 3: PHÁT TRIỂN ỨNG DỤNG WIKIPEDIA TRÊN IPHONE	36
3.1 Nhu cầu duyệt web và tra cứu thông tin trên thiết bị di động :	36
3.2.Mạng Wipipedia là gì :	37
3.3.Phát triển ứng dụng truy cập Wikipedia trên Iphone :	37
3.3.1.Mô tả ứng dụng :	37
3.3.2.Chức năng ứng dụng :	38
3.3.Môi trường làm việc :	43
3.4.Giao diện chương trình :	44
KẾT LUẬN	45

DANH MỤC CÁC HÌNH

Hình 1 : Kiến trúc hệ điều hành iOS

Hình 2 : Các loại region

Hình 3 : Region và Subregions

Hình 4 : Các trạng thái của một tiến trình iOS

Hình 5 : Sự phân mảnh bộ nhớ

Hình 6 : Kiến trúc hệ điều hành iPhone

Hình 7 : Các hàm thư viện của iOS

Hình 8 : Biểu đồ lượng người dùng truy cập Internet bằng điện thoại di động

MỞ ĐẦU

Dưới sự phát triển vượt bậc của Công nghệ thông tin, thời đại Internet bùng nổ chiếc điện thoại di động không chỉ đơn giản là phương tiện liên lạc mà nó còn là công cụ hữu ích cho con người.

Ngoài các chức năng nghe, gọi, nhắn tin thông thường, ngày nay điện thoại di động còn có rất nhiều ứng dụng khác như : quay phim, chụp ảnh, nghe nhạc, chơi game, gửi mail, tìm kiếm thông tin...

Wikipedia là một bách khoa toàn thư nội dung mở bằng nhiều ngôn ngữ trên Internet. Wikipedia được viết và xây dựng do rất nhiều người dùng cùng cộng tác với nhau. Việc xây dựng ứng dụng truy cập Wikipedia trên iPhone là một ứng dụng rất hữu ích.

Cùng với sự phát triển của khoa học công nghệ, chỉ đơn giản với một chiếc điện thoại di động và mạng Internet, người dùng có thể tìm kiếm thông tin mà không cần đến máy tính.

CHƯƠNG 1

KIẾN TRÚC HỆ ĐIỀU HÀNH iOS

1.1. Định nghĩa iOS :

iOS viết tắt của từ Internetwork Operating System, là một hệ điều hành hoạt động trên phần cứng của router Cisco, nó điều khiển hoạt động định tuyến và chuyển mạch của một router. Trên hệ điều hành iOS thì gồm có 3 phần : aaaa-bbbb-cccc trong đó :

- aaaa : dòng sản phẩm áp dụng hệ điều hành này
- bbbb : các tính năng của iOS
- cc : định dạng file iOS, nơi iOS chạy, kiểu nén của iOS.

Ví dụ : tên một iOS : C3620-i-mz_113-8T.bin.

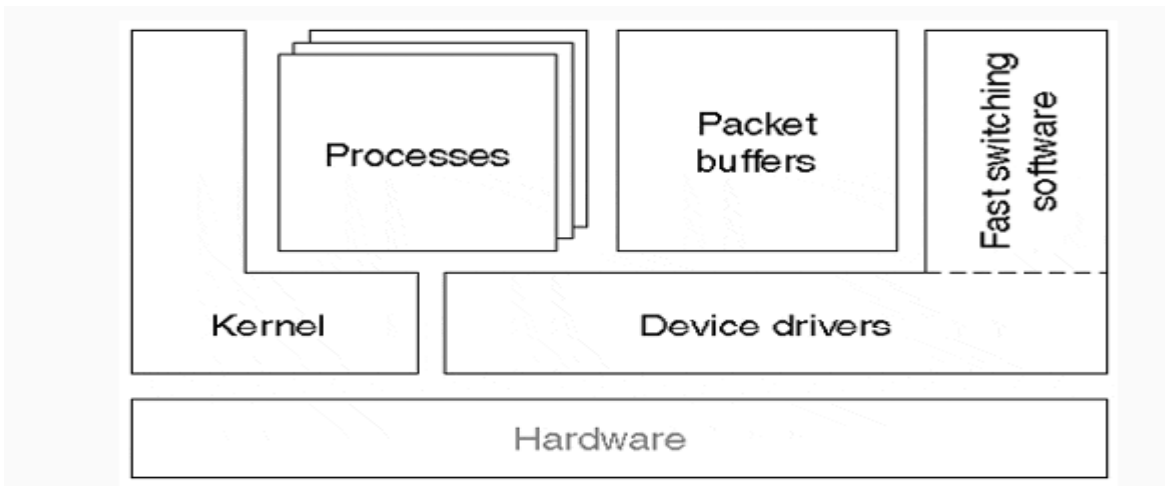
iOS là hệ điều hành trên các thiết bị di động của Apple. Ban đầu hệ điều hành này chỉ được phát triển để chạy trên iPhone, nhưng sau đó nó đã được mở rộng để chạy trên

các thiết bị của Apple như iPod touch, iPad và Apple TV.

Giao diện người dùng của iOS dựa trên cơ sở thao tác bằng tay trên màn hình cảm ứng của các thiết bị Apple.

1.2. Tổng quan về kiến trúc của iOS:

Khi mà lợi ích của việc định tuyến qua mạng trở nên phát triển, đòi hỏi router phải hỗ trợ một số những giao thức và cung cấp những chức năng khác, như cầu nối giữa các mạng. Cisco đã thêm những tính năng mới cho phần mềm của router. Kết quả có nhiều chức năng cầu nối và định tuyến như ngày nay. Nhưng hầu như cấu trúc cơ bản của hệ điều hành vẫn giống như ban đầu. iOS là một cấu trúc đơn giản, nhỏ, được thiết kế dựa vào những ràng buộc về bộ nhớ, về tốc độ, phần cứng của router.



Hình 1: Kiến trúc hệ điều hành iOS

Các thành phần của hệ điều hành iOS :

1.2.1 .Các tiến trình (Processes):

Là những tuyến riêng lẻ kết hợp với dữ liệu để thực hiện những tác vụ, như duy trì hệ thống, chuyển mạch gói dữ liệu, thực hiện giao thức định tuyến...

1.2.2.Nhân (Kernel):

Cung cấp những dịch vụ cơ bản của hệ thống tùy thuộc vào iOS như : quản lý bộ nhớ, lập lịch các tiến trình...Nó cung cấp quản lý tài nguyên phần cứng (CPU, bộ nhớ) cho các tiến trình.

1.2.3.Bộ đệm gói (Packet buffer):

Cung cấp các bộ đệm toàn cục và kết hợp với chức năng quản lý bộ đệm để lưu trữ gói dữ liệu đang được chuyển mạch.

1.2.4.Trình điều khiển thiết bị (Device driver):

Làm chức năng điều khiển giao tiếp giữa phần cứng và thiết bị ngoại vi, giao tiếp giữa các tiến trình iOS, kernel và phần cứng.Chúng cũng giao tiếp với phần mềm chuyển mạch nhanh (fast switching software).

1.2.5.Phần mềm chuyển mạch nhanh (Fast switching soft):

Chức năng chuyển mạch gói dữ liệu cao.

1.3.Tổ chức bộ nhớ:

iOS ánh xạ toàn bộ bộ nhớ vật lý thành một không gian địa chỉ ảo rộng lớn. MMU (Memory Map Unit) của CPU có giá trị khi được sử dụng để tạo không gian địa chỉ ảo thậm chí khi mà iOS không tận dụng một khối nhớ ảo trọn vẹn.

1.3.1. Miền bộ nhớ (Memory region) :

iOS chia không gian địa chỉ này thành những miền bộ nhớ gọi là region, mỗi region phù hợp với những loại bộ nhớ vật lý khác nhau.

Ví dụ : SRAM có thể lưu trữ gói dữ liệu và DRAM có thể lưu trữ phần mềm hoặc dữ liệu.

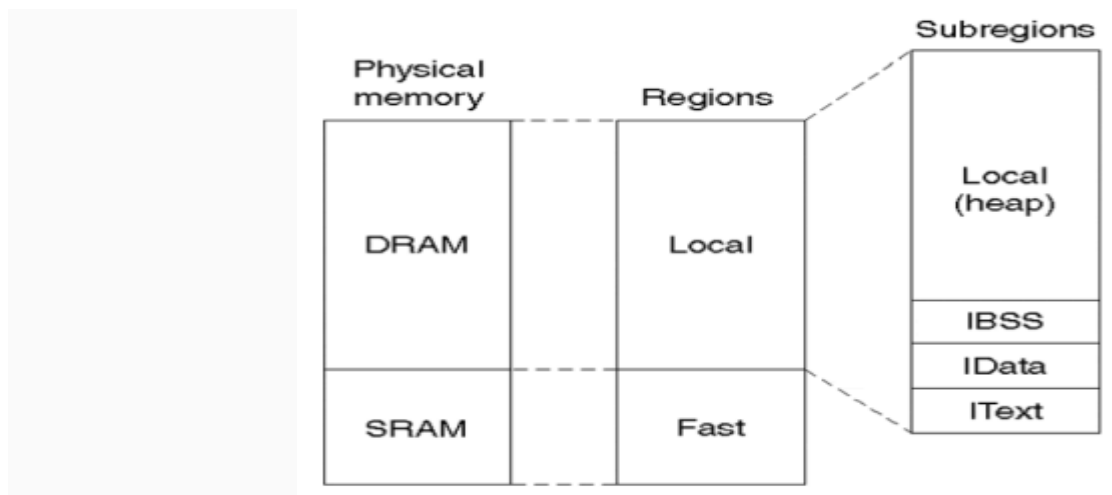
Phân lớp bộ nhớ thành các region cho phép iOS phân loại các bộ nhớ khác nhau vì vậy mà phần mềm không cần biết chi tiết về bộ nhớ trên mỗi platform .Các region được phân chia thành một trong tám mục như hình :

Memory region	Đặc điểm
Local(cục bộ)	Thông thường lưu trữ cấu trúc dữ liệu lúc chạy và local heap, thường là DRAM
Lomem	Bộ nhớ chia sẻ CPU và bộ điều khiển môi trường mạng sử dụng thông qua một bus dữ liệu, thường là SRAM
Fast	Bộ nhớ truy xuất nhanh, như SRAM, sử dụng cho mục đích đặc biệt và những tác vụ xem yêu cầu về tốc độ
Itext	Thực thi mã nguồn của iOS
Idata	Các biến được khởi tạo
IBss	Các biến không được khởi tạo
PCI	Bộ nhớ bus CPI, được sử dụng bởi tất cả thiết bị trên các bus PCI
Flash	Bộ nhớ flash dùng để lưu trữ iOS chạy từ RAM hoặc iOS chạy từ flash, nó cũng có thể lưu trữ một bảng file(tập tin) cấu hình dự phòng và những dữ liệu khác. Thông thường thì file hệ thống được xây dựng ở miền bộ nhớ flash này.

Hình 2: Các loại Region

1.3.2. Vùng bộ nhớ (Memory pool):

iOS quản lý bộ nhớ rồi thông qua một chuỗi các memory pool. Mỗi pool là một tập hợp các khối nhớ mà có thể cấp phát và thu hồi khi cần. Memory pool được xây dựng bên ngoài các region và được quản lý bởi kernel. Thường thì pool tương đương với một region đặc biệt.



Hình 3 : Region và Subregions

Một memory pool có thể xây dựng từ một vài region mở rộng, cho phép bộ nhớ được cấp phát và thu hồi từ các miền bộ nhớ khác nhau để tối đa hiệu quả hoạt động, có thể dùng lệnh `show memory` để hiển thị các pool này:

```
router#show memory Head Total(b) Used(b) Free(b) Lowest(b)
Largest(b) Processor 61281540 7858880 3314128 4544752 4377808 4485428
I/O1A00000 6291456 1326936 4964520 4951276 4964476 PCI 4B000000
1048576 407320 641256 6412556 641212 ...
```

Mô tả như sau(kích thước tính theo byte):

- Total: kích thước của pool.
- Used: lượng bộ nhớ được cấp phát.
- Free: lượng bộ nhớ sẵn sàng để sử dụng.
- Lowest: lượng bộ nhớ thấp nhất sẵn sàng sử dụng từ khi mà pool được tạo.
- Largest: kích thước khối nhớ liên tục lớn nhất sẵn sàng sử dụng hiện tại.

Lệnh `show memory` ở trên cung cấp ba pool với các tham số tương ứng: heap, processor và I/O.

1.4.Tiến trình iOS (Processes iOS):

1.4.1.Vòng đời của một tiến trình:

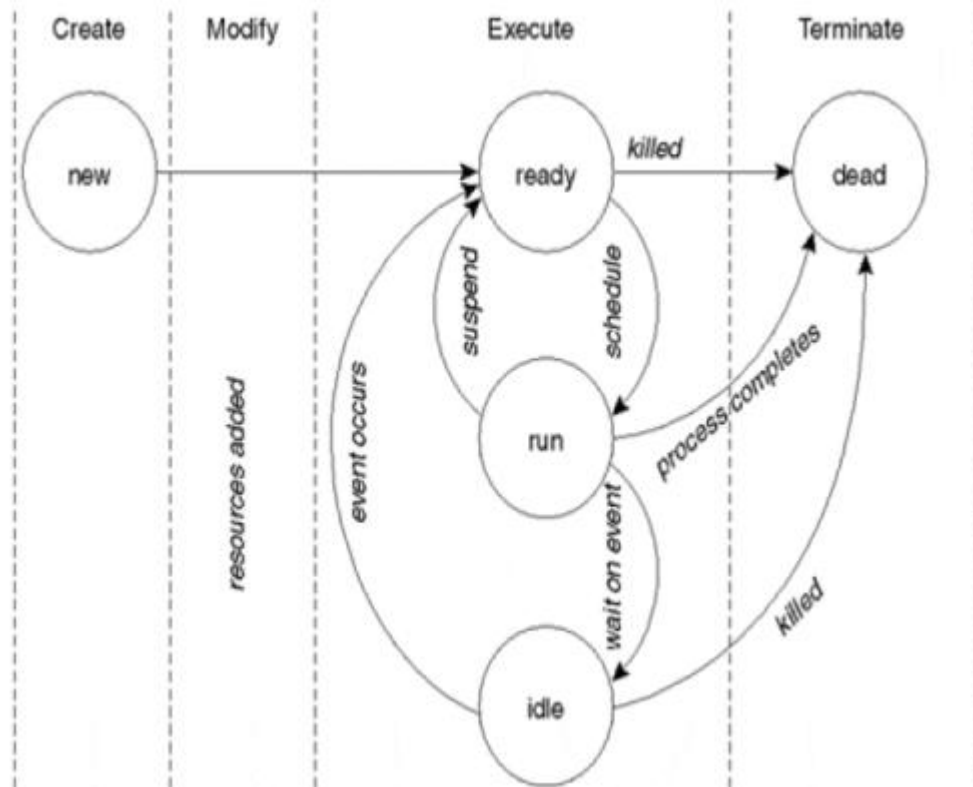
Một tiến trình có thể được tạo ra hoặc kết thúc bất cứ lúc nào trong khi iOS đang hoạt động ngoại trừ có ngắt xảy ra.Nó được tạo ra bởi kernel hoặc bởi một tiến trình khác đang chạy khác.

Một thành phần có trách nhiệm tạo nhiều tiến trình trong iOS gọi là parser(bộ phân tách).Parser này là một tập các chức năng làm phiên dịch cấu hình iOS và dòng lệnh EXEC.Parser được yêu cầu bởi kernel trong suốt quá trình khởi tạo iOS và các tiến trình EXEC, để cung cấp một giao tiếp dòng lệnh CLI thông qua giao tiếp console (giao tiếp người và máy) và các phiên telnet.Tại bất cứ thời điểm nào, một lệnh được nhập bởi người dùng hoặc một cấu hình được đọc từ file,parser phiên dịch dòng lệnh và đưa ra những hoạt động tức thời.Một vài lệnh cấu hình bởi việc gán trị, như địa chỉ IP, trong khi cấu hình khác như định tuyến hoặc giám sát. Một vài lệnh làm cho parser khởi tạo một tiến trình mới.

Ví dụ: khi mà lệnh cấu hình no router eigrp được nhập vào, parser khởi tạo một tiến trình mới, gọi là ipigrp (nếu như tiến trình ipigrp đã được khởi tạo rồi), bắt đầu xử lý gói ip.

EIGRP.Nếu như lệnh cấu hình no router eigrp được nhập vào, parser kết thúc tiến trình ipigrpvà không còn chức năng định tuyến EIGRP.

Tiến trình iOS trải qua các trạng thái như sau:



Hình 4 : Các trạng thái của một tiến trình iOS

1.4.1.1. Trạng thái khởi tạo (Create):

Khi mà một tiến trình mới được tạo, nó nhận vùng stack riêng của mình và vào trạng thái mới (new). Tiến trình có thể di chuyển đến trạng thái điều chỉnh (Modify). Nếu không có thay đổi cần thiết, thì tiến trình chuyển sang trạng thái thực thi (Execute).

1.4.1.2. Trạng thái điều chỉnh (Modify):

Không giống như hầu hết các hệ điều hành, iOS không tự động truyền tải các tham số khởi tạo hoặc gán một giao tiếp đến một tiến trình mới khi nó được tạo, bởi vì nó cho rằng hầu hết các tiến trình không cần tài nguyên này. Nếu một tiến trình cần nguồn tài nguyên này, tuyến mà tạo nó có thể điều chỉnh để thêm vào.

1.4.1.3. Trạng thái thực thi (Execute):

Sau khi một tiến trình mới được tạo thành công và điều chỉnh, nó chuyển sang trạng thái sẵn sàng (Ready) và vào trạng thái thực thi (Execute). Trong suốt trạng thái này, một tiến trình có thể truy cập CPU và chạy. Trong suốt trạng thái thực thi, một tiến trình có thể truy cập CPU và chạy. Trong suốt trạng thái thực

thi , một tiến trình có thể là một trong 3 trạng thái: sẵn sàng, chạy và rỗi (Idle). Một tiến trình ở trạng thái sẵn sàng sẽ đợi chuyển sang trạng thái truy cập CPU và bắt đầu thực thi lệnh. Một tiến trình ở trạng thái rỗi là đang ngủ, đợi sự kiện bên ngoài xuất hiện trước khi nó có thể chạy. Một tiến trình chuyển từ trạng thái sẵn sàng sang trạng thái chạy khi mà nó được lập lịch để chạy.

Với đa tác vụ mà không ưu tiên (non-preemptive multitasking), một tiến trình được lập lịch chạy trên CPU cho đến khi tạm ngừng hoặc kết thúc. Một tiến trình có thể tạm dừng theo 2 cách: nó có thể tự dừng bởi việc báo cho kernel, nó muốn nhường cho CPU và chuyển sang trạng thái sẵn sàng, và đợi đến lúc chạy lại. Tiến trình cũng có thể dừng bởi một hoạt động bên ngoài xảy ra. Khi mà một tiến trình đợi một sự kiện, kernel hoàn toàn dừng tiến trình này và chuyển nó sang trạng thái rỗi. Sau khi một sự kiện xảy ra rồi thì kernel chuyển tiến trình trở lại trạng thái sẵn sàng để đợi chạy lại.

1.4.1.4. Trạng thái kết thúc (Terminal):

Trạng thái cuối cùng trong vòng đời của tiến trình là trạng thái kết thúc. Một tiến trình vào trạng thái kết thúc khi nó hoàn thành chức năng của mình và đóng lại hoặc khi một tiến trình khác đóng nó. Khi một tiến trình bị đóng hoặc tự đóng, tiến trình chuyển sang trạng thái chết (Dead). Tiến trình này ở trạng thái chết (không hoạt động) cho đến khi kernel thu hồi tất cả các tài nguyên của nó. Sau khi tài nguyên được thu hồi, tiến trình bị kết thúc thoát khỏi trạng thái chết và xóa khỏi hệ thống.

1.4.2. Độ ưu tiên tiến trình iOS:

iOS thực hiện chế độ ưu tiên để lập lịch các tiến trình trên CPU. Tại thời điểm tạo, mỗi tiến trình được gán một trong 4 độ ưu tiên dựa trên mục đích của tiến trình. Độ ưu tiên là không đổi, chúng được gán khi một tiến trình được tạo và không bao giờ thay đổi. Các độ ưu tiên:

-Critical:

Dành riêng cho những tiến trình hệ thống thiết yếu mà giải quyết những vấn đề cấp phát tài nguyên.

-High:

Được gán cho những tiến trình mà cung cấp đáp ứng nhanh, như tiến trình nhận gói trực tiếp từ giao tiếp mạng .

-Medium:

Độ ưu tiên mặc định sử dụng bởi hầu hết các tiến trình.

-Low:

Được gán cho những tiến trình cung cấp những tác vụ mang tính định kỳ như bảng ghi lỗi... Độ ưu tiên các tiến trình cung cấp sự ưu đãi cho một vài tiến trình để truy cập CPU dựa trên sự quan trọng của nó đối với hệ thống và iOS không thực hiện quyền ưu tiên. Một tiến trình có sự ưu tiên cao hơn không thể ngắt một tiến trình có độ ưu tiên thấp hơn, thay vào đó, tiến trình có độ ưu tiên cao hơn thì có nhiều cơ hội hơn để truy cập CPU hơn.

1.5. Kernel iOS:

iOS Kernel không là một đơn vị mà là một tập các thành phần và chức năng liên kết chặt chẽ với nhau. iOS Kernel thực hiện các chức năng sau: Lập lịch tiến trình, quản lý bộ nhớ, cung cấp dịch vụ retimes để trap (phát hiện) và handle (điều khiển) những ngắt phần cứng, duy trì timer (bộ định thời gian), và phát hiện ngoại lệ phần mềm.

Các chức năng chính của Kernel:

1.5.1. Lập lịch:

Tác vụ lập lịch các tiến trình được thực hiện bởi scheduler (bộ lập lịch).

Scheduler quản lý tất cả các tiến trình trong hệ thống bằng cách sử dụng một chuỗi các hàng đợi tiến trình mô tả trạng thái của mỗi tiến trình. Các hàng đợi này chứa nội dung thông tin cho tiến trình ở trạng thái đó. Tiến trình chuyển từ trạng thái này sang trạng thái khác khi mà scheduler di chuyển ngữ cảnh từ 1 hàng đợi tiến trình này đến hàng đợi tiến trình khác.

1.5.2. Quản lý bộ nhớ:

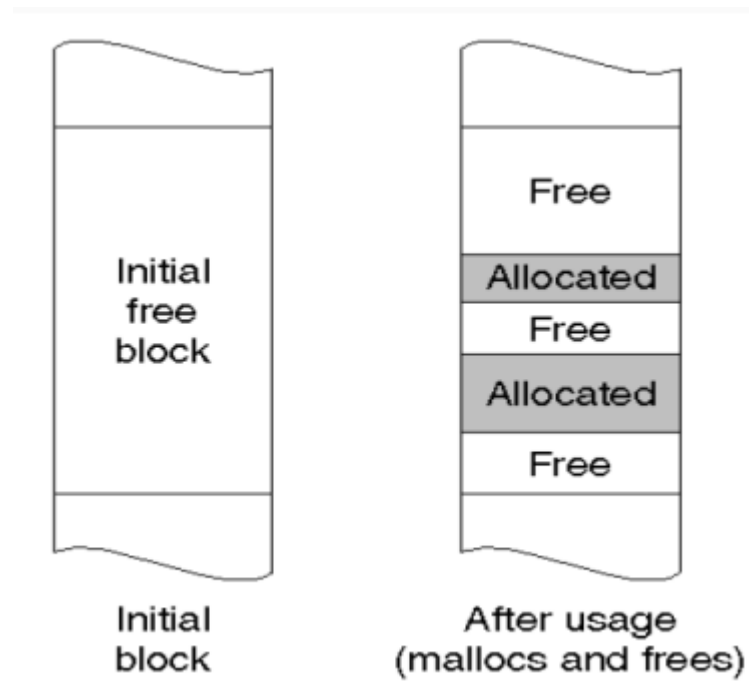
Bộ quản lý bộ nhớ của Kernel tại mức quá cao quản lý tất cả các vùng nhớ có sẵn của iOS, bao gồm bộ nhớ chứa iOS của nó. Bộ quản lý bộ nhớ ba thành phần riêng biệt, với những nhiệm vụ riêng.

Có ba bộ quản lý bộ nhớ sau:

Bộ quản lý Region, Bộ quản lý Pool, Bộ quản lý Chunk.

1.5.2.1. Bộ quản lý Region:

Định nghĩa và duy trì những region khác nhau trên một platform. Bộ quản lý region có chức năng duy trì tất cả các region. Nó cung cấp các dịch vụ cho phép những phần khác của iOS tạo region và gán các thuộc tính của chúng. Nó cũng cho phép những phần khác truy vấn những region có sẵn, ví dụ quyết định tổng lượng bộ nhớ có sẵn trên một platform.



Hình 5: Sự phân mảnh bộ nhớ

1.5.2.2. Bộ quản lý pool:

Quản lý việc tạo ra các vùng nhớ pool, cấp phát và thu hồi các khối nhớ của pool.

Bộ quản lý pool là một thành phần quan trọng của Kernel. Trong khi scheduler quản lý cấp phát tài nguyên CPU để xử lý, bộ quản lý pool cấp phát bộ nhớ cho các tiến trình. Tất cả các tiến trình phải thông qua bộ quản lý bộ nhớ pool trực tiếp hoặc gián tiếp, để định ra vùng nhớ mà nó sử dụng. Bộ quản lý bộ nhớ được yêu cầu cho mỗi tiến trình sử dụng hàm hệ thống chuẩn malloc và free để lấy và trả bộ nhớ. Bộ quản lý bộ nhớ hoạt động bởi việc duy trì danh sách khối nhớ rỗi cho mỗi pool, ban đầu mỗi pool chứa chỉ một khối nhớ rỗi lớn bằng

kích thước một pool. Khi bộ quản lý bộ nhớ pool yêu cầu bộ nhớ, khởi tạo những khối nhớ có kích thước nhỏ hơn. Tại cùng một thời điểm, các tiến trình có thể giải phóng vùng nhớ trả về pool, tạo thành một số vùng nhớ rời không liên tục nhau, nhiều kích thước, trường hợp này gọi là phân mảnh bộ nhớ.

1.5.2.2. Quản lý Chunk:

Quản lý pool cung cấp nhiều cách hiệu quả để quản lý một tập hợp các khối có kích thước khác nhau. Tuy nhiên những tính năng này có chi phí của nó, bộ quản lý pool tạo ra 32 byte overhead trên mỗi bộ nhớ. Mặc dù overhead này không quan trọng lắm đối với khối dữ liệu lớn, đối với pool có hàng ngàn khối dữ liệu nhỏ hơn thì overhead mới trở nên đáng quan tâm. Để tạo thêm sự lựa chọn thì kernel cung cấp bộ quản lý bộ nhớ khác gọi là bộ quản lý bộ nhớ Chunk, nó có thể quản lý lượng lớn pool có nhiều khối nhớ nhỏ mà không có overhead. Không giống như quản lý pool, bộ quản lý Chunk không tạo ra danh sách vùng nhớ rời với kích thước khác nhau. Thay vào đó bộ quản lý chunk quản lý một tập các khối nhớ cố định được chỉ định từ một trong các vùng nhớ pool. Trong một vài trường hợp, bộ quản lý chunk có thể xem như là một bộ quản lý pool vùng nhớ con.

Các chính sách thường được thực hiện là:

Một tiến trình yêu cầu một vị trí của mooth khối nhớ từ một vùng nhớ pool đặc biệt. Một tiến trình sau đó gọi đến bộ quản lý chunk để chia khối nhớ thành một chuỗi các chunk có kích thước cố định và nhỏ hơn. Sử dụng bộ quản lý chunk để định vị ra những vùng nhớ rời khi cần. Thuận lợi là ở đây chỉ có 32 byte overhead và bộ quản lý pool thì không bắt buộc cấp phát và lấy lại hàng ngàn phân mảnh nhỏ hơn. Vì vậy khả năng phân mảnh trong pool giảm đáng kể.

1.5.3. Quản lý bộ đệm gói:

Trong định tuyến gói dữ liệu, bất cứ hoạt động lưu trữ hay chuyển dữ liệu đều phải cần có một nơi để lưu trữ dữ liệu trong khi dữ liệu đang được định tuyến trên đường truyền. Thông thường thì tạo ra một bộ đệm để lưu trữ các gói đến trong khi hoạt động chuyển mạch đang hoạt động. Bởi vì khả năng định tuyến gói là trung tâm của cấu trúc hệ điều hành iOS. iOS chứa thành phần

chuyên biệt để quản lý những vùng đệm này. Thành phần này được gọi là bộ quản lý vùng đệm bộ nhớ. iOS sử dụng thành phần này để tạo và quản lý nhất quán một chuỗi các vùng đệm cho chuyên mạch trên mỗi platform. Bộ đệm trong vùng này được biết chung là những bộ đệm hệ thống. Bộ quản lý vùng bộ đệm cung cấp một cách tiện lợi để quản lý một tập các bộ đệm có kích thước cụ thể. Mặt khác nó có thể được sử dụng để quản lý bất cứ loại bộ đệm nào, bộ quản lý bộ đệm được sử dụng chính để quản lý những vùng bộ đệm gói. Những vùng đệm gói được tạo từ các vùng nhớ pool có sẵn. Để tạo một vùng, bộ quản lý bộ đệm yêu cầu một khối nhớ từ bộ quản lý pool và chia cho bộ đệm. Bộ quản lý bộ đệm gói sau đó tạo một danh sách cho tất cả các bộ đệm rồi và theo dõi các vùng nhớ này. Những vùng đệm có thể là động hay tĩnh, vùng bộ đệm tĩnh được tạo với số bộ đệm cố định, không thêm bộ đệm vào cùng bộ đệm tĩnh này. Vùng bộ đệm động được tạo với một số bộ đệm tối thiểu. Gọi là bộ đệm thường xuyên, có thể thêm hoặc xóa các bộ đệm. Với các vùng nhớ động, nếu bộ quản lý bộ đệm nhận được yêu cầu khi vùng nhớ rỗng, nó cố gắng mở rộng vùng nhớ và đáp ứng yêu cầu ngay lập tức.

Nếu nó không thể mở rộng vùng nhớ, thì yêu cầu bị lỗi và thực hiện mở rộng vùng nhớ sau đó. vùng bộ đệm được phân lớp public (dùng chung) hoặc private (dùng riêng).

Vùng public được sử dụng bởi bất cứ tiến trình nào của hệ thống.

Private được tạo cho một tập các tiến trình sử dụng.

*Bộ đệm hệ thống:

Mỗi iOS đều có một tập các bộ đệm public định trước gọi là những bộ đệm hệ thống, những bộ đệm này được sử dụng cho tiến trình chuyên mạch các gói dữ liệu và tạo gói (như gói keepalive, gói cập nhật định tuyến).

1.6. Trình điều khiển thiết bị:

iOS chứa trình điều khiển thiết bị cho các thiết bị phần cứng, như flash card, NVRAM, nhưng đáng chú ý là trình điều khiển cho các giao tiếp mạng. Trình điều khiển các giao tiếp mạng cung cấp những khả năng chính cho hoạt động của gói dữ liệu tại đầu ra của giao tiếp.

Mỗi thiết bị chứa hai thành phần chính: thành phần điều khiển và thành phần dữ liệu.

Thành phần điều khiển có trách nhiệm quản lý tình trạng và trạng thái của thiết bị (ví dụ: shutdown trên một cổng).

Thành phần dữ liệu có trách nhiệm đối với tất cả các luồng hoạt động chuyển mạch gói dữ liệu.

Trình điều khiển thiết bị có quan hệ chặt chẽ với chức năng chuyển mạch gói.

Trình điều khiển thiết bị giao tiếp mạng dựa trên hệ thống iOS thông qua một cấu trúc điều khiển đặc biệt gọi là IDB (interface descriptor block).

IDB chứa toàn bộ chức năng điều khiển thiết bị, dữ liệu, trạng thái thiết bị.

Ví dụ: địa chỉ IP, trạng thái cổng, thông kê gói là một trong các trường hiện tại trong IDB.iOS duy trì một IDB cho mỗi giao tiếp hiện tại trên một platform.

*Cấu trúc chuyển mạch gói:

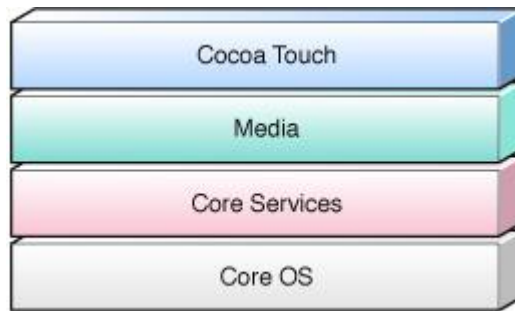
Chuyển mạch gói là một trong các chức năng quan trọng nhất của router, hoạt động bao quát như sau:

- Một gói đến một cổng.
- Địa chỉ đích của gói được kiểm tra và so sánh dựa vào danh sách đích đã biết.
- Nếu phù hợp, gói được chuyển tiếp ra cổng phù hợp.
- Nếu không phù hợp, gói bị hủy.Rõ ràng vấn đề chuyển mạch không quá phức tạp, nhưng để chuyển mạch được nhanh thì vấn đề trở nên phức tạp.Tốc độ hoạt động không chỉ dựa vào tốc độ CPU, còn có những nhân tố khác, khả năng thực thi của bus I/O. tốc độ bộ nhớ đều có ảnh hưởng đến sự thực thi của chuyển mạch.Đây là một thử thách lớn đối với các nhà phát triển iOS.Làm thế nào để tốc độ chuyển mạch nhanh nhất mà có thể giới hạn trên các thành phần CPU, bộ nhớ, bus I/O.Khi mà kích thước và số mạng định tuyến tăng lên, những người phát triển iOS tiếp tục tìm ra những cách tốt nhất để giải quyết thử thách thực thi này.

Đầu tiên thì iOS chỉ có chức năng chuyển mạch nhưng đã được cải thiện về sau, một vài phương thức chuyển mạch dựa trên nhiều platform khác nhau.Ngày nay iOS có thể chuyển mạch tới vài trăm ngàn gói trên một giây, sử dụng bạn định tuyến tới vài trăm ngàn tuyến đi.

1.7.Kiến trúc của hệ điều hành iPhone :

Bao gồm các lớp sau :



Hình 6 : Kiến trúc hệ điều hành iPhone

Các lớp dưới cùng là nền tảng của hệ điều hành, phụ trách quản lý bộ nhớ, các file hệ thống, mạng, các hệ điều hành nhiệm vụ và tương tác trực tiếp với các phần cứng.

1.7.1.Lớp Core OS :

Lớp Core OS bao gồm các thành phần :

- OS X Kernel
- Match 3.0
- BSD
- Sockets
- Security
- Power Management
- Key chain
- Certificates
- File System
- Bonjour

1.7.2.Lớp Core Services :

Lớp Core Services cung cấp một trừu tượng trên các dịch vụ được cung cấp trong lớp OS X Kernel.Nó cung cấp truy cập cơ bản để các dịch vụ hệ điều hành iPhone và bao gồm các thành phần sau :

- Collection
- AddressBook
- Networking
- Files Access

- SQLite
- Core Location
- Net Services
- Threading
- Preferences
- URL Utilities

1.7.3.Lớp Media(Truyền thông) :

Lớp Media cung cấp các dịch vụ đa phương tiện mà bạn có thể sử dụng trong iPhone và iPad.Nó bao gồm các thành phần sau :

- Core Audio
- OpenGL
- Audio Mixing
- Audio Recording
- Video Playback
- JPG,PNG,TIFF
- PDF
- Quartz
- Core Animation
- OpenGL ES

1.7.4.Lớp Cocoa Touch :

Lớp Coscoa Touch cung cấp một lớp trừu tượng để khai báo các thư viện khác nhau cho các lập trình iPhone và iPad, như sau:

- Multi-Touchcontrols
- Accelerometer
- View Hierachy
- Localization
- Alert
- Web View
- People Picker
- Image Picker
- Controllers.

CHƯƠNG 2:

PHÁT TRIỂN ỨNG DỤNG THI HÀNH TRÊN iOS

Để có thể viết ứng dụng trên iPhone chúng ta cần bộ công cụ phát triển iPhone SDK. Bao gồm :

- * Xcode : Môi trường phát triển tích hợp (IDE) cho phép người dùng quản lý, chỉnh sửa, và gỡ lỗi các dự án.

- * iPhone Simulator : Cung cấp một mô phỏng phần mềm để mô phỏng một chiếc iPhone hoặc iPad trên máy Mac của người dùng.

- * Instruments: Phân tích công cụ để giúp tối ưu hóa ứng dụng và màn hình của người dùng có bị rò rỉ bộ nhớ trong thời gian thực hay không.

- * iOS Framework : các hàm thư viện để thao tác với thiết bị : âm thanh, hình ảnh, GPS, cảm biến.

2.1. Lập trình Xcode :

Xcode là bộ công cụ hoàn chỉnh để xây dựng OS X và các ứng dụng iOS và với Xcode 4, các công cụ đã được thiết kế lại để chạy nhanh hơn, dễ dàng sử dụng và hữu ích hơn rất nhiều.

Bộ công cụ Xcode bao gồm : môi trường phát triển tích hợp (IDE), giao diện thiết kế (Builder), trình biên dịch (Apple LLVM).

Để khởi động Xcode, kích đúp vào biểu tượng của Xcode nằm trong thư mục /Developer/Applications



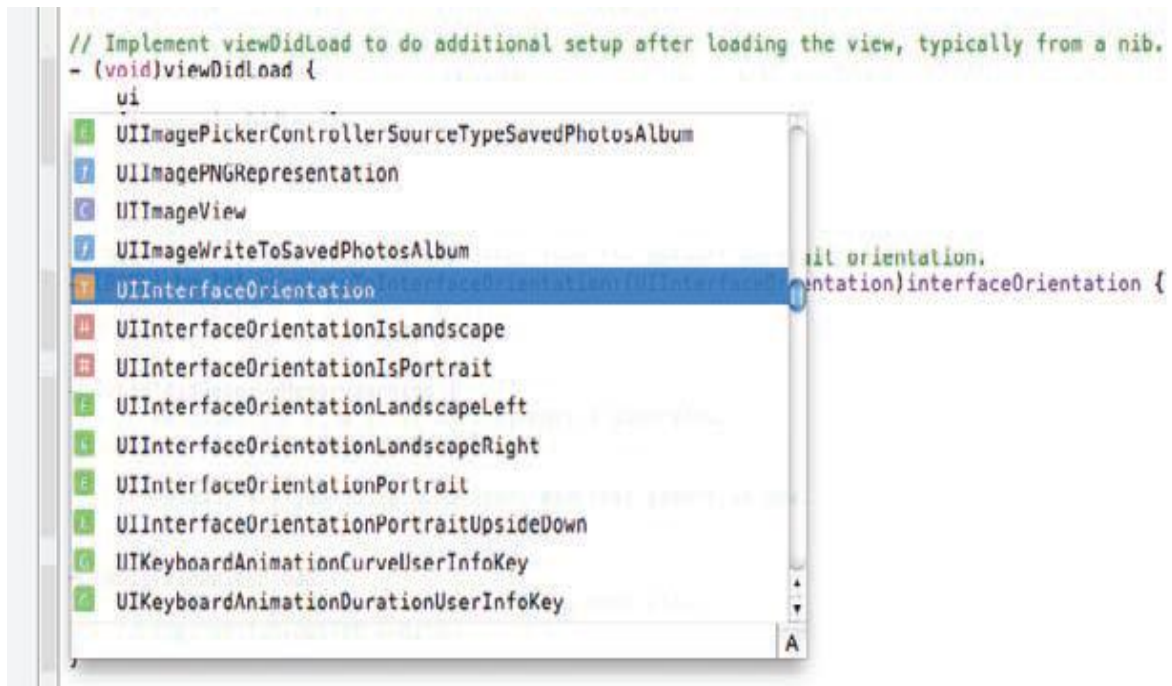
2.1.1.Xcode IDE (Môi trường phát triển tích hợp) :

Được thiết kế dựa trên các công nghệ mới nhất của Apple.Xcode IDE tích hợp tất cả các công cụ cần thiết để làm việc.Suốt quá trình chuyển đổi từ giao diện,soạn mã nguồn,gỡ lỗi đều được thực hiện trong cùng 1 cửa sổ.

Trong quá trình soạn thảo khi kiểm tra giá trị của 1 biến trong quá trình chạy.

IDE Xcode xác định những sai lầm trong cả hai cú pháp và logic, và thậm chí sẽ sửa chữa mã.

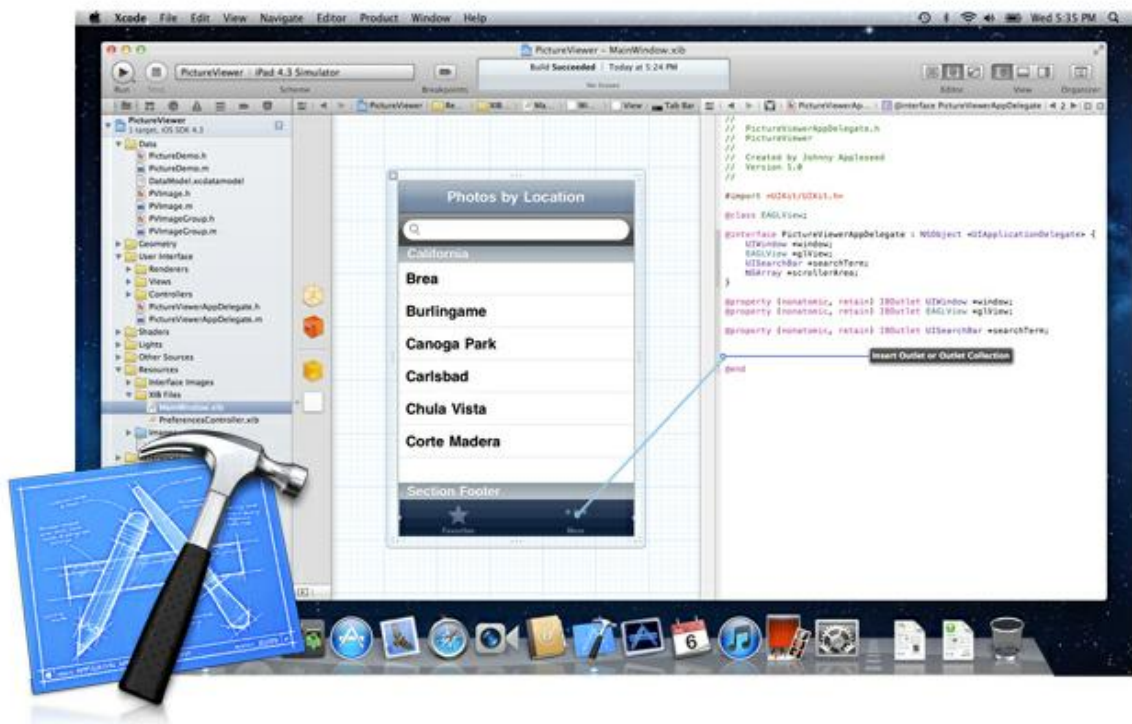
IDE trong Xcode cung cấp nhiều công cụ và tính năng giúp người dùng dễ dàng lập trình hơn nhiều.Một trong những tính năng là mã Sense hiển thị một danh sách popup, hiển thị các lớp và các thành viên sẵn có, chẳng hạn như phương pháp, thuộc tính...



2.1.1.1. Single Window (Cửa sổ đơn) :

Với Xcode 4 các cửa sổ sử dụng để thực hiện các nhiệm vụ được hợp nhất thành 1 cửa sổ duy nhất.

Xcode 4 với giao diện người dùng độc đáo, có thể dễ dàng làm việc trên nhiều nhiệm vụ khác nhau, thậm chí nhiều dự án mà không làm lộn xộn.



2.1.1.2.Navigators :

Navigators bao gồm danh sách các tập tin trong dự án, các biểu tượng, giao diện tìm kiếm người dùng để dàng sử dụng.



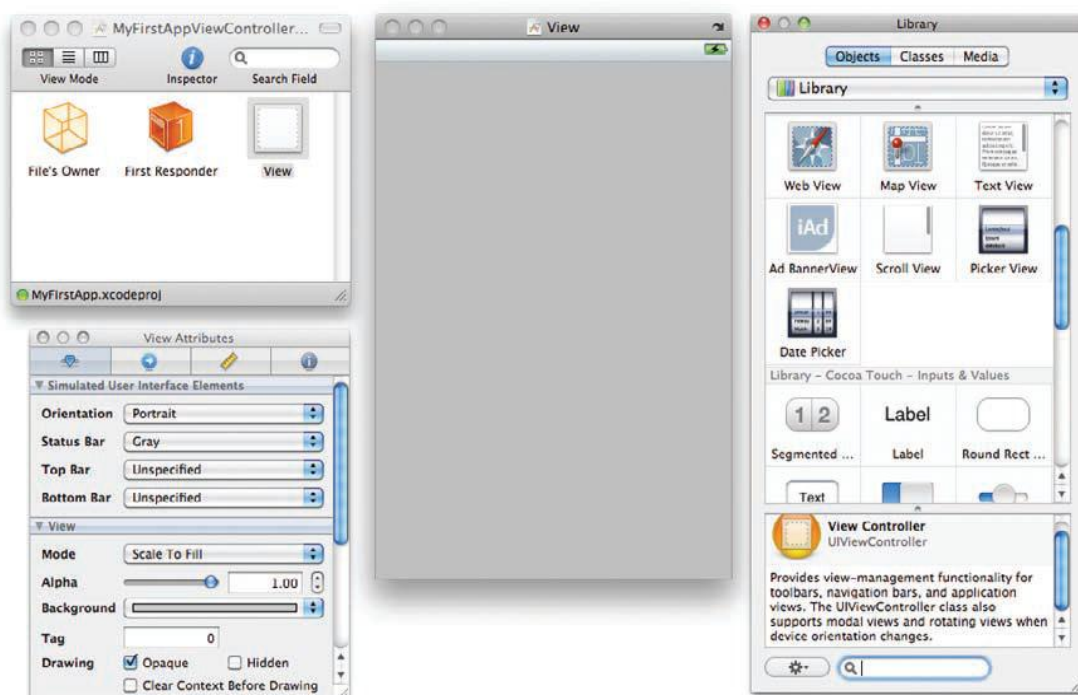
2.1.1.3.Jump Bar :



Ở phía trên của tất cả các cửa sổ soạn thảo là một thanh đường dẫn hiển thị vị trí tương đối của các tập tin hiện tại. Nhấp vào bất kỳ vị trí nào trong thanh Jump Bar thì sẽ chuyển đến file đó.

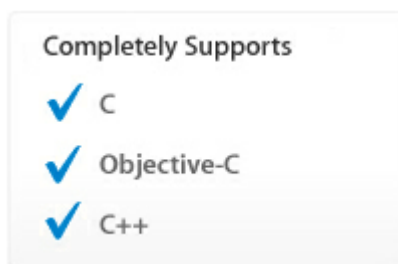
2.1.2.Interface Builder(Giao diện chương trình):

Giao diện Builder là một công cụ trực quan cho phép người dùng thiết kế giao diện cho các ứng dụng iPhone / iPad. Sử dụng giao diện Builder, kéo và thả lên cửa sổ và sau đó kết nối các cửa sổ khác nhau với các hành động để có thể lập trình tương tác với mã của chương trình

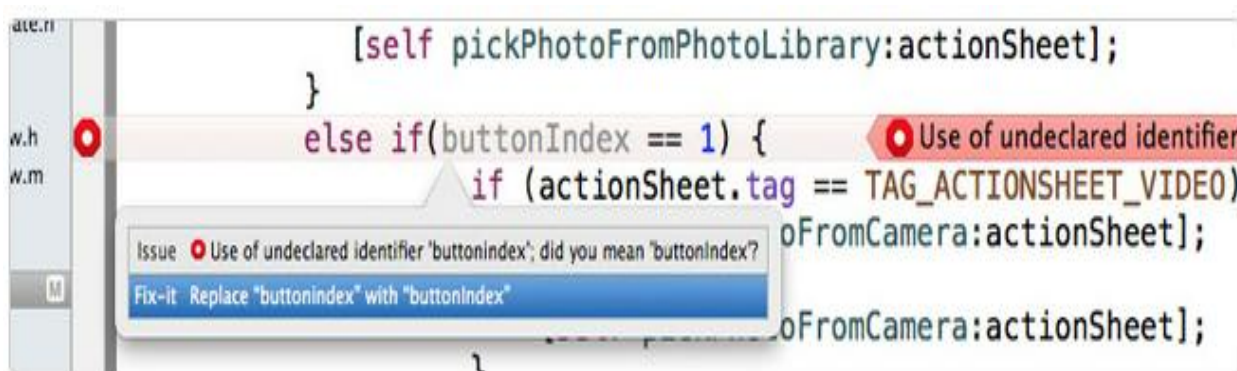


2.1.3.Apple LLVM(Trình biên dịch) :

Trình biên dịch trong Xcode bao gồm 1 tập hợp các thư viện được tối ưu hóa,dễ dàng mở rộng.Trong Xcode 4, trình biên dịch stack - từ phân tích cú pháp trước rồi tối ưu hóa mã hoàn toàn hỗ trợ cho C, Objective-C, C ++.



Xcode IDE còn có tính năng sửa chữa các lỗi tự động.Ngoài việc báo cáo lỗi, IDE rất thông minh,trong nhiều trường hợp Xcode sẽ không chỉ báo cáo một lỗi nó sẽ trình bày một giải pháp tốt để khắc phục bằng cách nhấp chuột vào lỗi để sửa chữa.Ví dụ như sửa chữa một biểu tượng sai chính tả hoặc thêm một dấu chấm phẩy còn thiếu...Một phím tắt ngay lập tức sẽ có lỗi sửa chữa và cho phép người dùng tiếp tục mã hóa.



Sử dụng Xcode có thể phát triển các loại khác nhau của iPhone, iPad, và các ứng dụng Mac OS X.(Không phải tất cả các mẫu ứng dụng có sẵn cho iPhone và iPad.Ví dụ, dựa trên mẫu Navigation-ứng dụng không hỗ trợ iPad, Split-View ứng dụng không hỗ trợ iPhone).

2.2. Iphone Simulator(Mô phỏng Iphone) :

Simulator iPhone là một công cụ rất hữu ích mà có thể sử dụng để thử nghiệm ứng dụng của mà không cần sử dụng iPhone / iPod touch / iPad..Simulator iPhone được đặt tại thư mục

/Developer/Platforms/iPhoneSimulator.platform/Developer/Applications.

Xcode tự động cài đặt các ứng dụng trên iPhone Simulator.

Simulator iPhone có thể mô phỏng các phiên bản khác nhau của hệ điều hành iPhone



Các tính năng của iPhone Simulator :

Simulator iPhone mô phỏng các tính năng khác nhau của một chiếc iPhone thực sự như iPod touch, hoặc thiết bị iPad. Các tính năng có thể thử nghiệm trên Simulator iPhone bao gồm :

* Xoay màn hình : bên trái, bên phải, trên đầu.

* Hỗ trợ cho hành động :

- Tap

- Touch and Hold

- Double-tap

- Swipe

- Flick

- Drag

- Pinch

Tuy nhiên cũng có 1 số tính năng không có sẵn trên iPhone Simulator :

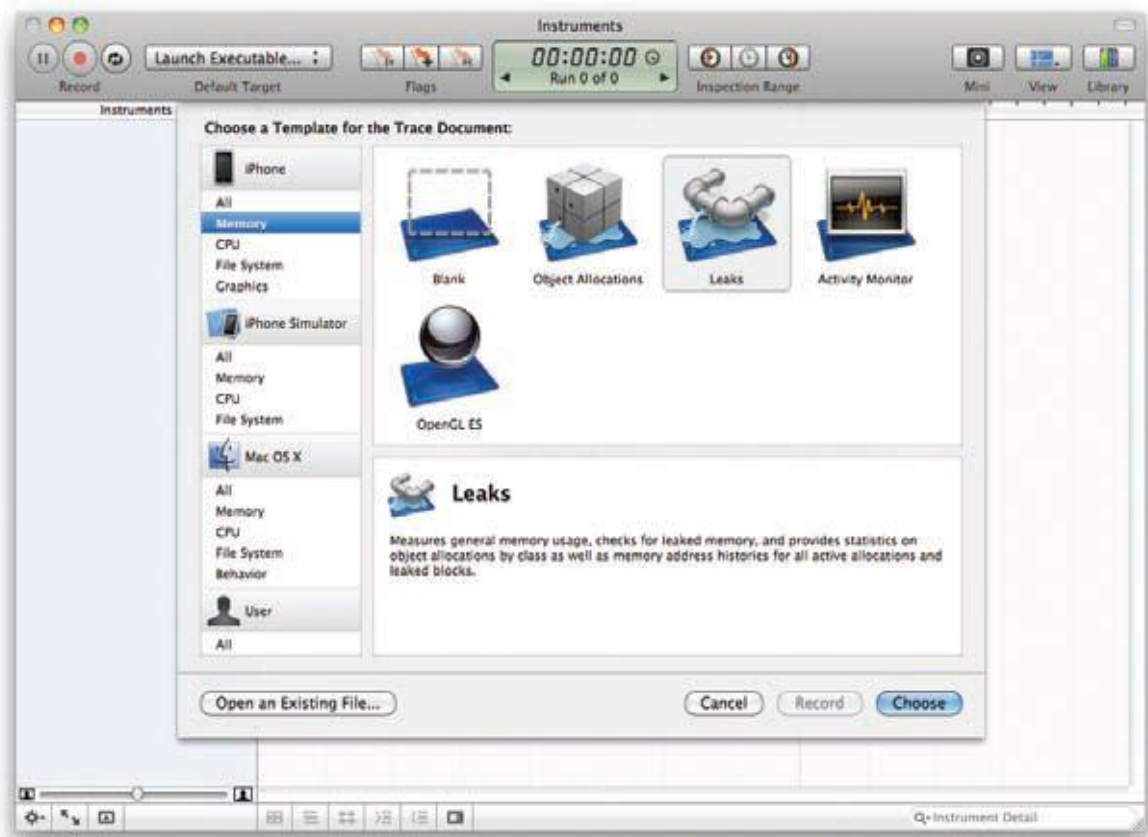
- Gọi điện thoại

- Truy cập các gia tốc

- Gửi và nhận tin nhắn
- Cài đặt ứng dụng từ App Store
- Máy ảnh
- Microphone
- Một số tính năng của OpenGL ES

2. 3.Instrument :

Các ứng dụng cho phép truy cập tự động và theo dõi các hồ sơ thực hiện của hệ điều hành Mac OS X, iPhone, và các ứng dụng iPad.



2.4. iOS Framework :

Các hàm thư viện của iOS :

Tên	Tiền tố	Mô tả
Accelerate.framework	cblas,vDSP	Thông tin toán học
Accounts.framework	AC	Có giao diện để quản lý truy cập vào tài khoản của người sử dụng một hệ thống

AdressBook.framework	AB	Chứa các chức năng để truy cập vào CSDL danh bạ của người dùng trực tiếp
AdressBookUI.framework	AB	Chứa các lớp để hiển thị hệ thống xác định người dùng và các giao diện trình soạn thảo
AssetsLibrary.framework	AL	Chứa các lớp truy cập hình ảnh và video của người dùng
AudioToolbox.framework	AU,Audio	Chứa các giao diện để xử lý các dòng dữ liệu âm thanh
AudioUnit.framework	AU,Audio	Chứa các giao diện tải và đơn vị sử dụng âm thanh
AVFoundation.framework	AV	Chứa giao diện để chơi và ghi lại âm thanh,video
CFNetwork.framework	CF	Chứa giao diện để truy cập mạng thông qua Wi-fi và radio di động
CoreAudio.framework	Audio	Cung cấp các loại dữ liệu được sử dụng trong cả âm thanh Core
CoreBluetooth.framework	CB	Cung cấp sự truy cập phần cứng Bluetooth năng lượng thấp
CoreData.framework	NS	Thông tin giao diện quản lý ứng dụng mô hình dữ liệu
CoreFoundation.framework	CF	Cung cấp các dịch vụ phần mềm cơ bản,bao gồm khái niệm trừu tượng cho các loại dữ liệu phổ biến, chuỗi tiện ích, tiện ích thu thập, quản lý tài nguyên

CoreGraphics.framework	CG	Chứa các giao diện cho Quartz 2D
CoreImage.framework	CI	Chứa giao diện để thao tác với video và hình ảnh tĩnh
CoreLocation.framework	CL	Chứa các giao diện để xác định vị trí người dùng
CoreMedia.framework	CM	Chứa giao diện để thao tác với âm thanh và video
CoreMIDI.framework	MIDI	Chứa giao diện để xử lý dữ liệu MIDI
CoreMotion.framework	CM	Chứa giao diện để truy cập dữ liệu tốc độ
CoreTelephony.framework	CT	Chứa giao diện truy cập thông tin liên quan đến điện thoại
CoreText.framework	CT	Chứa giao diện thao tác với văn bản và vẽ
CoreVideo.framework	CV	Chứa giao diện cho các thao tác âm thanh và video
EventKit.framework	EK	Chứa giao diện để truy cập dữ liệu sự kiện lịch của người dùng
EventKitUI.framework	EK	Chứa các lớp để hiển thị giao diện hệ thống lịch tiêu chuẩn
ExternalAccessory.framework	EA	Chứa giao diện để giao tiếp với các phụ kiện phần cứng
Foundation.framework	NS	Chứa giao diện cho các chuỗi quản lý, bộ sưu tập và các loại dữ liệu khác ở mức độ thấp
GameKit.framework	GK	Chứa các giao diện để quản lý kết nối peer-to-peer

GLKit.framework	GLK	Chứa các lớp Objective-C tiện ích để xây dựng các ứng dụng OpenGL ES phức tạp
GSS.framework	gss	Cung cấp một bộ tiêu chuẩn các dịch vụ liên quan đến an ninh
iAd.framework	AD	Chứa các giao diện để hiển thị các quảng cáo trong ứng dụng
ImageIO.framework	CG	Chứa các giao diện để đọc và ghi dữ liệu hình ảnh
IIOKit.framework	N/A	Chứa các giao diện được sử dụng bởi thiết bị
MapKit.framework	MK	Chứa các giao diện để nhúng một giao diện bản đồ vào ứng dụng và cho tọa độ mã hóa địa lý đảo ngược
MediaPlayer.framework	MP	Chứa giao diện để chơi video và xem toàn màn hình
MessageUI.framework	MF	Chứa giao diện để viết tin nhắn và email
MobielCoreService.framework	UT	Xác định định danh được hỗ trợ bởi hệ thống
NewsstandKit.framework	NK	Cung cấp các giao diện để tải về nội dung tạp chí và tờ báo ở chế độ nền
OpenAL.framework	AL	Chứa các giao diện cho OpenAL, thư viện âm thanh
OpenGLES.framework	EAGL, GL	Chứa các giao diện OpenGL ES, là một phiên bản nhúng của các nền tảng OpenGL 2D và 3D, thư viện đồ họa

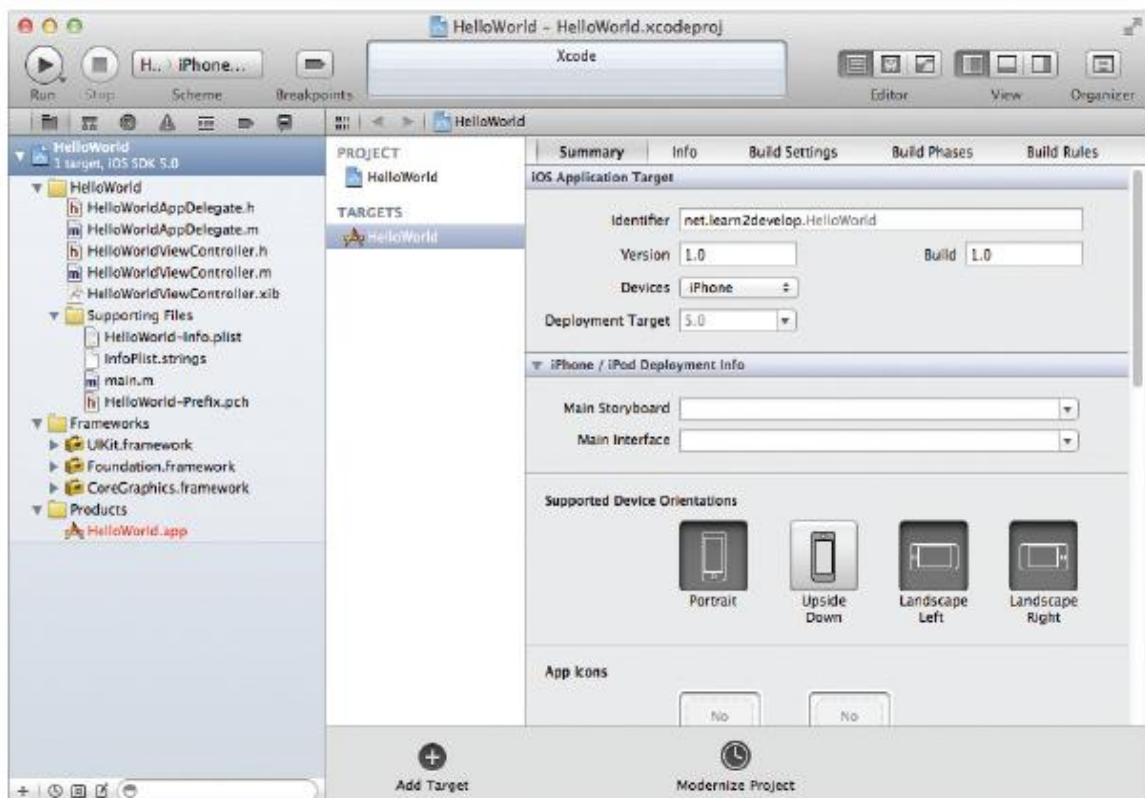
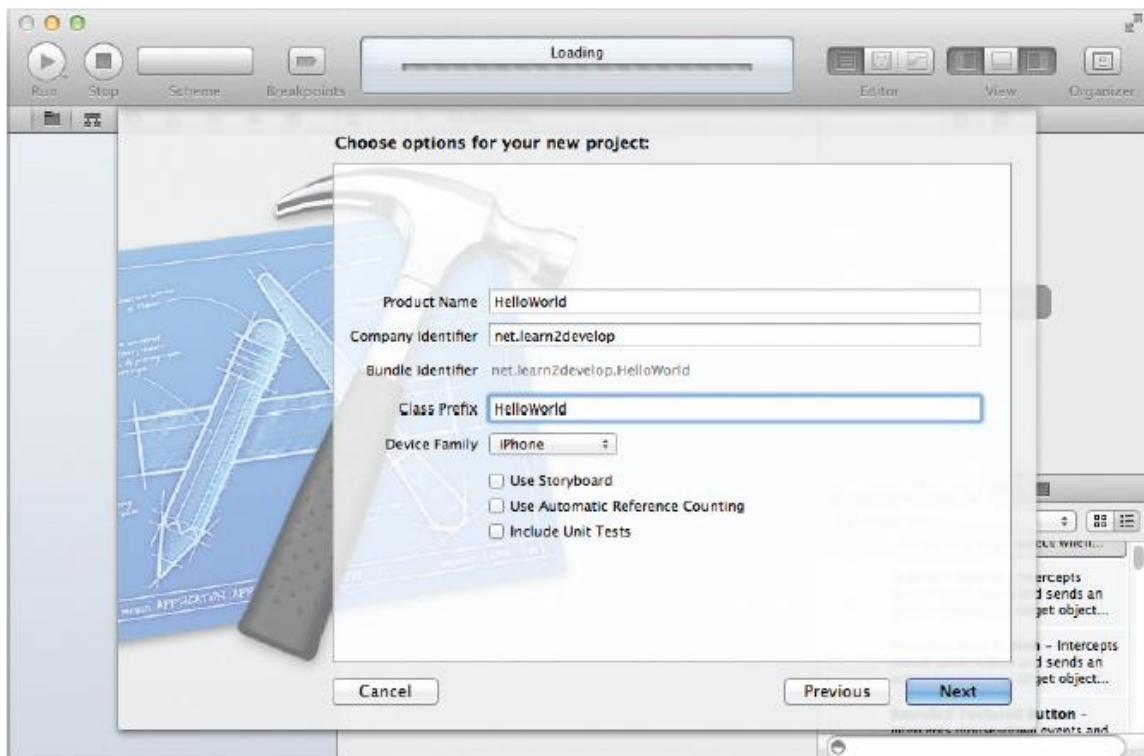
QuartzCore.framework	CA	Chứa giao diện Core Animation
QuickLook.framework	QL	Chứa giao diện cho các tập tin xem trước
Security.framework	CSSM,Sec	Chứa giao diện cho các chứng chỉ quản lý, bảo mật
StoreKit.framework	SK	Chứa giao diện để xử lý các giao dịch tài chính liên quan đến các ứng dụng mua hàng
SytemConfiguration.framework	SC	Chứa giao diện để xác định cấu hình mạng của một thiết bị
Twitter.framework	TW	Chứa giao diện để gửi tweets thông qua dịch vụ Twitter
UIKir.framework	UI	Chứa các lớp và phương pháp cho các lớp ứng dụng giao diện người sử dụng iOS

Hình 7 : Các hàm thư viện của iOS

2.5. Một số ứng dụng đơn giản :

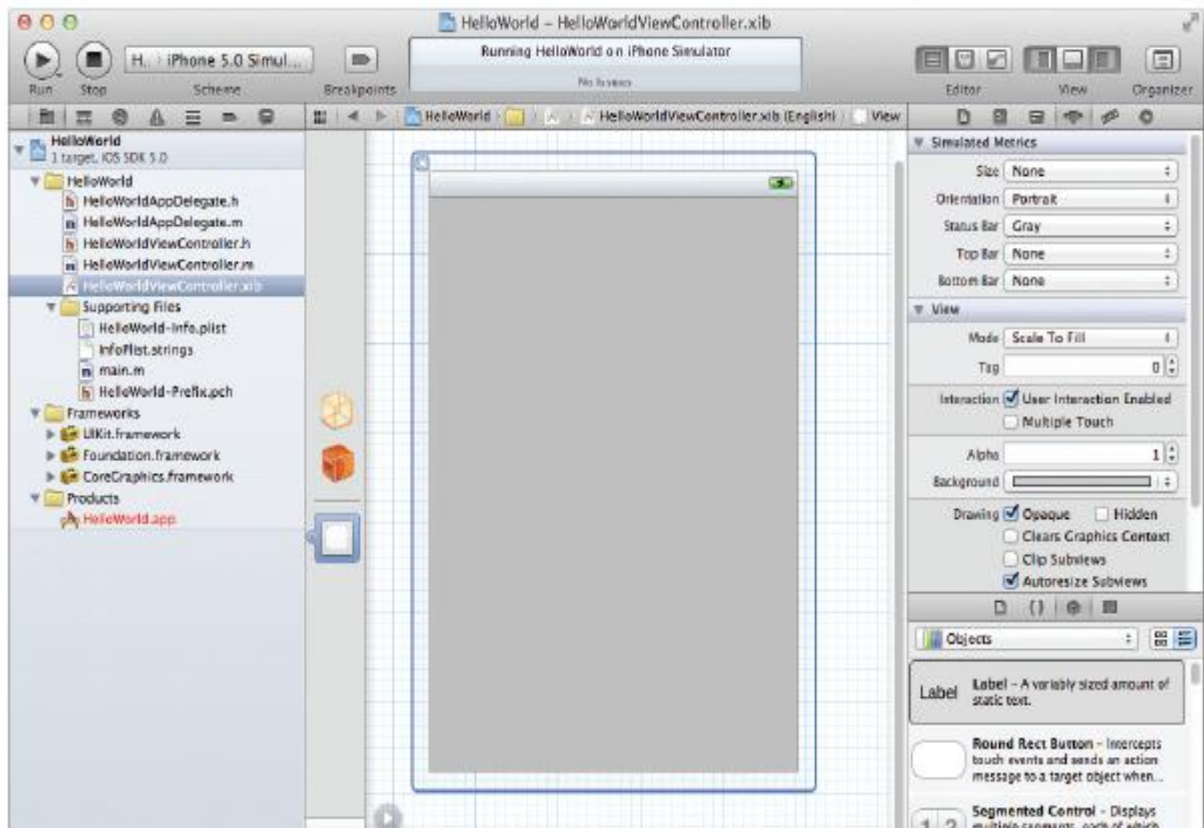
2.5.1. Chương trình Hello World :

- Chạy Xcode, đặt tên project là HelloWorld

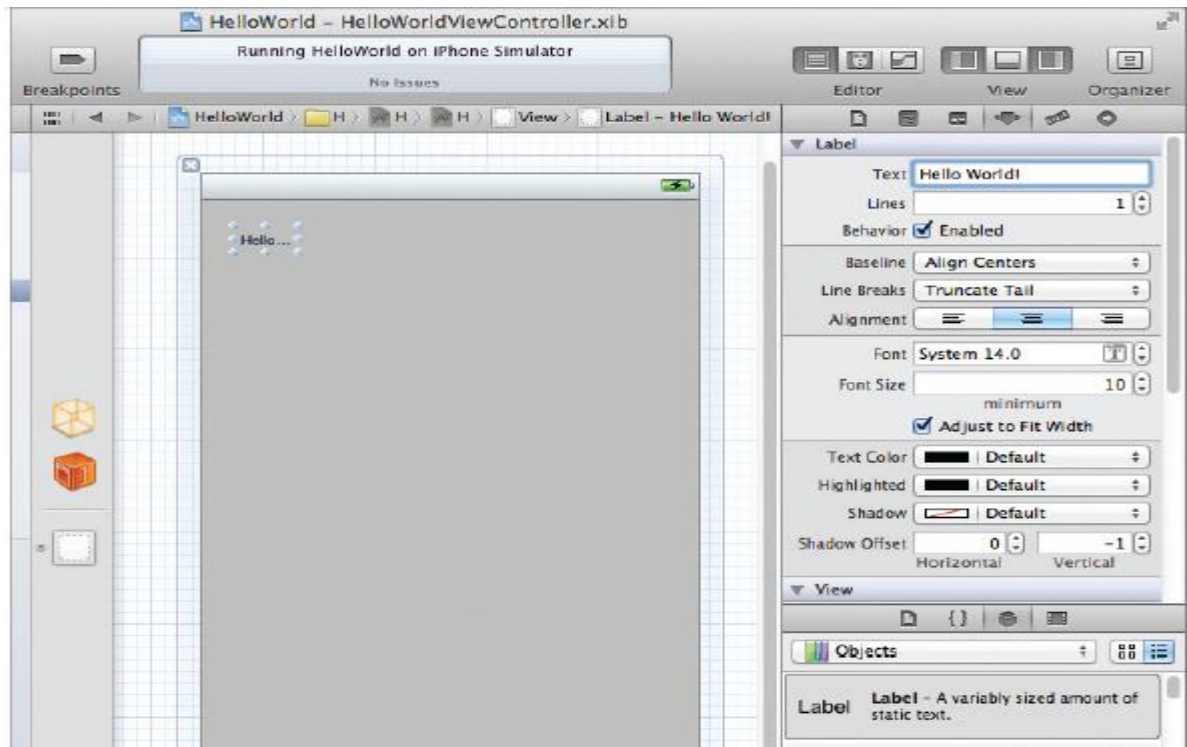


*Thiết kế giao diện :

Chọn tập HelloWorldViewController.xib để chỉnh sửa nó bằng cách sử dụng giao diện Builder



Trong khu vực Utilities trên bên phải, phần Thư viện đối tượng và di chuyển xuống để xem Label và kéo và thả một Label, Text field, Round Rect Button vào cửa sổ View



*Code :

Trong tập tin HelloWorldViewController.h, thêm một khai báo hành động btnClicked:

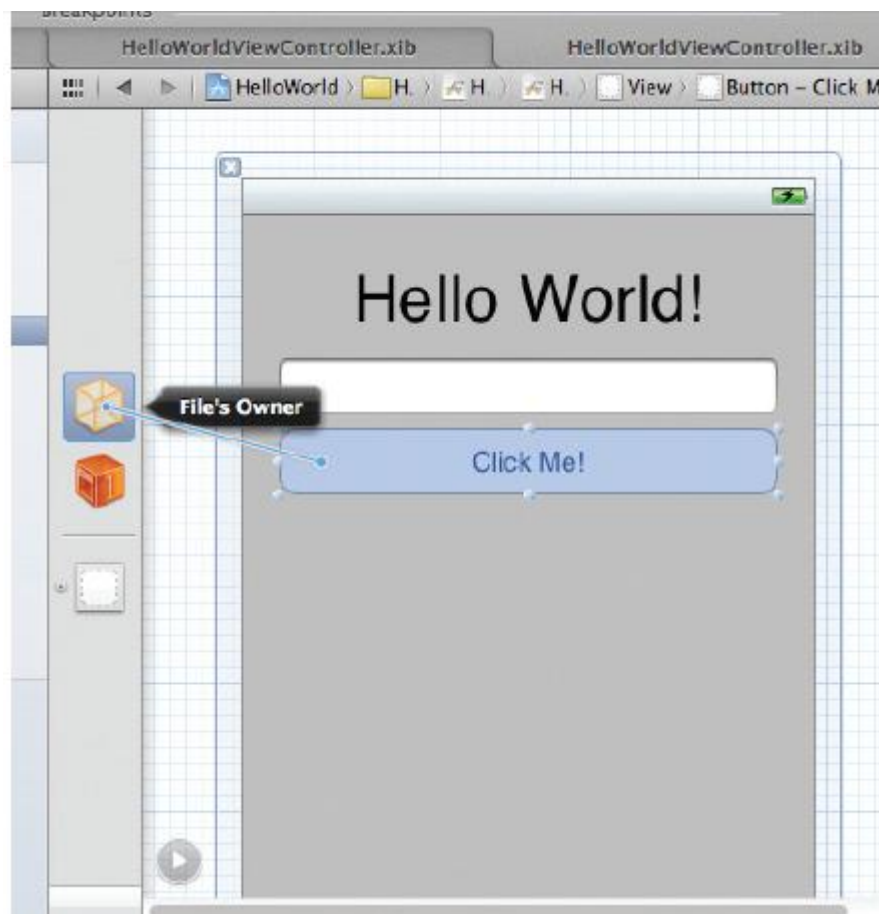
```
#import <UIKit/UIKit.h>
```

```
@interface HelloWorldViewController : UIViewController
```

```
-(IBAction) btnClicked:(id) sender;
```

```
@end
```

Quay lại tập tin HelloWorldViewController.xib click chuột phải kéo thả btnClicked vào File's Owner



Trong tập tin HelloWorldViewController.m thêm các lệnh :

```
#import "HelloWorldViewController.h"
@implementation HelloWorldViewController
-(IBAction) btnClicked:(id) sender {
//---display an alert view---
UIAlertView *alert =
[[UIAlertView alloc] initWithTitle:@"Hello World!"
message:@"iPhone, here I come!"
delegate:self
cancelButtonTitle:@"OK"
otherButtonTitles:nil];
[alert show];
[alert release];
}
```

Ấn tổ hợp phím Ctrl+R chạy chương trình



2.5.2. Đóng gói chương trình:

+Bước 1:

Vào Menu chọn Window -> Organizer -> Devices Section -> Provisioning Profiles

-> Under Library group -> Chọn Automatic Device Provisioning.

+Bước 2:

Chọn Refresh -> nhập Apple ID

+Bước 3:

Click vào tên Project -> chọn Project and Produce Setting -> Appropriate profile -> chọn iPhone Distribution -> iPhone 5.0 Simulator.

+Bước 4:

Vào Product -> Archive -> Next -> Next -> đổi tên file thành .ipa

CHƯƠNG 3

PHÁT TRIỂN ỨNG DỤNG WIKIPEDIA TRÊN IPHONE

3.1 Nhu cầu duyệt web và tra cứu thông tin trên thiết bị di động :

Trong vòng 10 năm trở lại đây, tỷ lệ sử dụng Internet tại Việt Nam có tốc độ tăng trưởng nhanh nhất trong khu vực. Việt Nam lọt vào top 20 nước sử dụng Internet nhiều nhất trên thế giới (theo báo cáo “Hiện trạng Internet toàn cầu” do Pingdom – hãng chuyên cung cấp dịch vụ giám sát hoạt động của website, máy chủ cho tổ chức, doanh nghiệp).

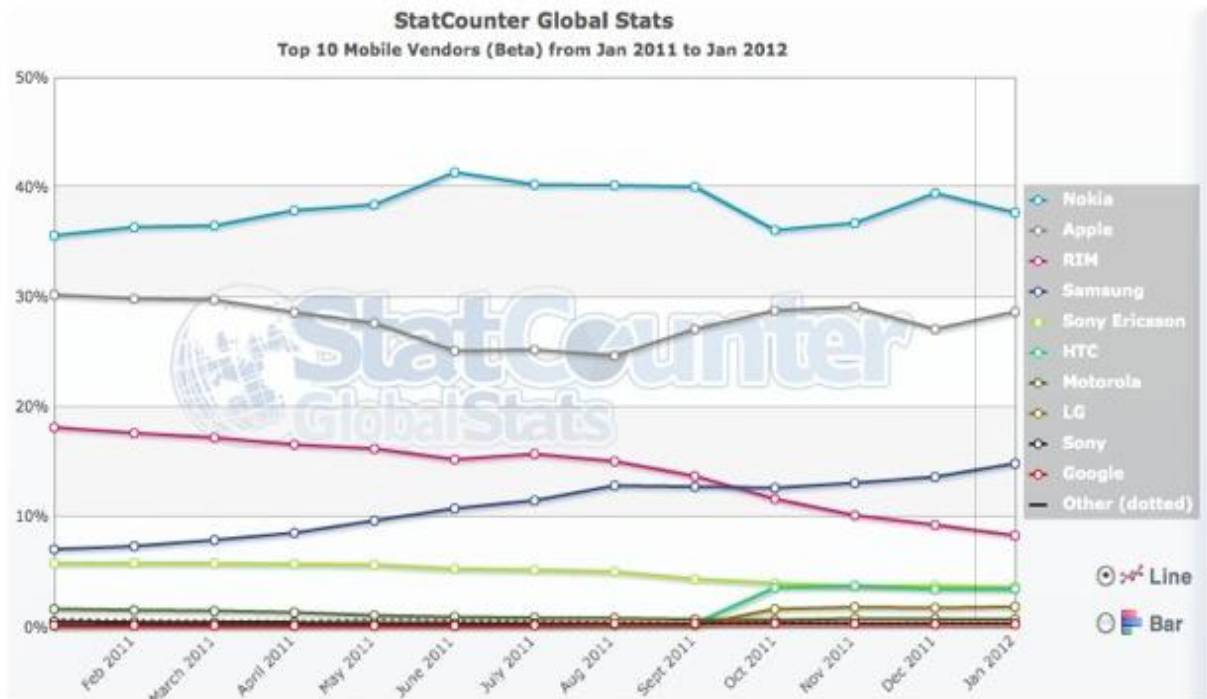
Đặc biệt Việt Nam là quốc gia có tốc độ phát triển Internet di động nhanh thứ 2 trong khu vực Đông Nam Á (chiếm 60%), chỉ đứng sau Malaysia.

Dựa trên những con số chính thức, 31% dân số Việt Nam có truy cập Internet và mỗi năm có thêm 2-3 triệu người truy cập Internet.

Internet được truy cập rất thường xuyên. Hầu hết tất cả người sử dụng Internet truy cập Internet hàng ngày hoặc nhiều lần một tuần. Một nghiên cứu với 5.800 người tại 12 thành phố lớn, kết quả là so với năm 2010 thì trong năm 2011 số người dùng các thiết bị truy cập vào Internet đã thay đổi. Cụ thể, người dùng sử dụng máy tính để bàn truy cập vào Internet đã giảm từ 84% trong năm 2010 xuống còn 81% trong năm 2011. Trong khi đó, tỷ lệ người dùng laptop truy cập Internet tăng từ 38% lên 47%, đặc biệt số người sử dụng điện thoại di động truy cập vào Internet đã tăng hơn gấp đôi – từ 27% năm 2010 lên 56% trong năm 2011.

Thống kê từ StatCounter cho thấy lượng người dùng truy cập vào Internet từ điện thoại di động đang tăng với tốc độ ổn định sau mỗi năm.

Trong tổng số điện thoại truy cập Internet trong năm vừa qua người dùng tại Mỹ và Anh truy cập vào Internet từ Iphone là nhiều nhất.



Hình 8 : Biểu đồ lượng người dùng truy cập Internet bằng điện thoại di động

3.2. Mạng Wikipedia là gì :

Wikipedia là một bách khoa toàn thư nội dung mở bằng nhiều ngôn ngữ trên Internet. Wikipedia được viết và xây dựng do rất nhiều người dùng cùng cộng tác với nhau.

Hiện nay Wikipedia trực thuộc Quỹ hỗ trợ Wikimedia, một tổ chức phi lợi nhuận. Wikipedia hiện có hơn 15 triệu bài viết, với hơn 3,3 triệu bài trong phiên bản tiếng Anh (English Wikipedia). Wikipedia hiện có hơn 200 phiên bản ngôn ngữ, trong đó vào khoảng 100 đang hoạt động. 15 phiên bản đã có hơn 50.000 bài viết : tiếng Anh, Đức, Pháp, Ba Lan, Nhật, Ý, Thụy Điển, Hà Lan, Bồ Đào Nha, Tây Ban Nha, Nga, Na Uy, Phần Lan và tiếng Việt.

Wikipedia là một tài liệu tham khảo mang tính bách khoa, không phải là một cẩm nang hướng dẫn sử dụng, sách du lịch, hoặc sách giáo khoa.

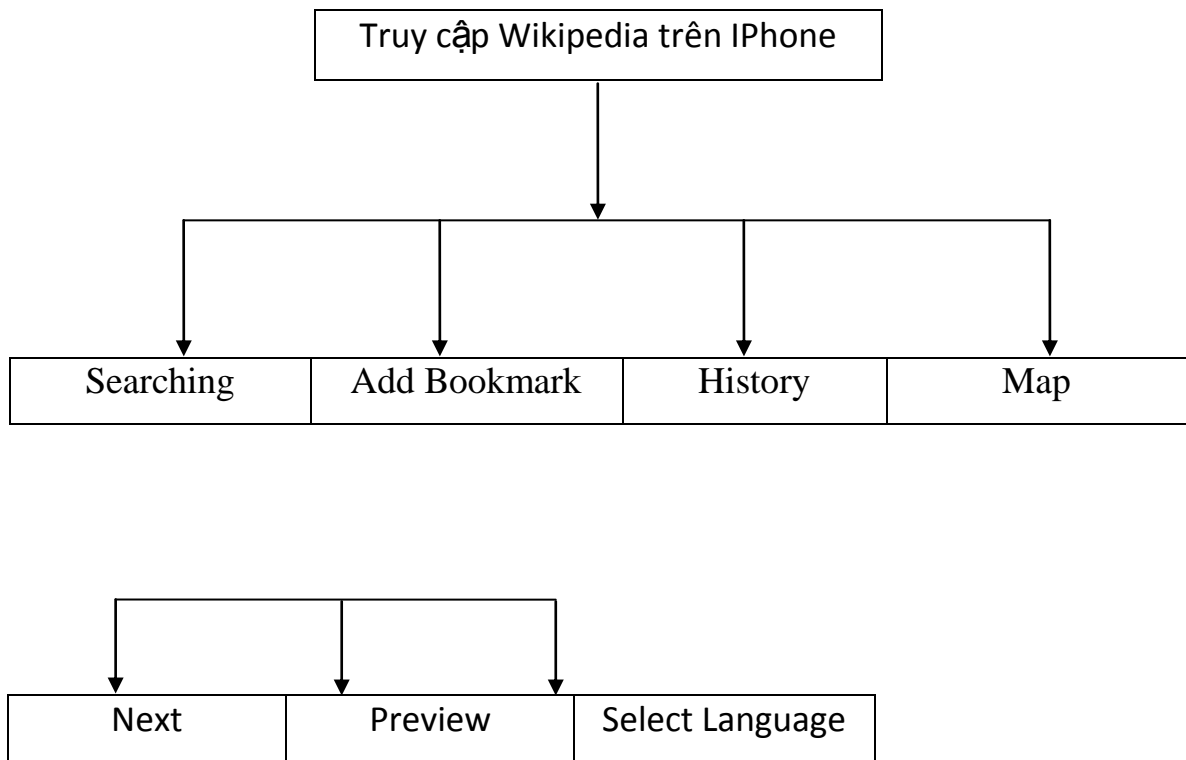
3.3. Phát triển ứng dụng truy cập Wikipedia trên Iphone :

3.3.1. Mô tả ứng dụng :

Ứng dụng Wikipedia trên Iphone cho phép truy cập bất kì bài viết nào của Wikipedia với giao diện đơn giản, gọn gàng với ô tìm kiếm nằm trên cùng, bên dưới là bài viết nổi bật được hiển thị ngẫu nhiên.

3.3.2.Chức năng ứng dụng :

Wikipedia trên Iphone với các tính năng : Searching, Add Bookmark, History, Map, Next, Preview, Select Language.



Hình 8 : Các chức năng của ứng dụng

- Searching : chức năng tìm kiếm thông tin.

Chức năng này cho phép tìm kiếm các thông tin trên Wikipedia.

+ Sử dụng lớp thư viện : Foundation.h

+ Code :

```
- (void)searchBar:(UISearchBar *)searchBar textDidChange:(NSString *)searchText  
{  
    searchText = [searchText  
stringByAddingPercentEscapesUsingEncoding:NSUTF8StringEncoding];  
    NSString *urlString = [NSString  
stringWithFormat:@"http://%@.wikipedia.org/w/api.php?action=opensearch&search  
=%@&format=json", [appDelegate.settings valueForKey:@"languageKey"],  
searchText];  
    NSURL *url = [NSURL URLWithString:urlString];  
    NSURLRequest *request = [[NSURLRequest alloc] initWithURL:url];
```

```

        NSURLConnection *connection = [[NSURLConnection alloc]
initWithRequest:request delegate:self];
        [connection release];
        [request release];
        if ([searchText length] > 0)
        {
            tableView.alpha = 1.0;
            tableView.hidden = NO;
        }
        else
        {
            tableView.alpha = 0.0;
            tableView.hidden = YES;
        }
        [tableView setContentOffset:CGPointMake(0, 0) animated:YES];
    }
- (void)connection:(NSURLConnection *)connection
didReceiveResponse:(NSURLResponse *)response
{
}
- (void)connection:(NSURLConnection *)connection didReceiveData:(NSData *)data
{
    NSString *jsonString = [[NSString alloc] initWithData:data
encoding:NSUTF8StringEncoding];
    NSArray *results = [jsonString JSONValue];

    if (results && [results count] >= 1)
    {
        searchResults = [NSMutableArray arrayWithArray:[results objectAtIndex:1]];
    } else
    {
        searchResults = [NSMutableArray array];
    }
}

```



```

    [searchResults retain];
    [jsonString release];
    [tableView reloadData];
}

```

```

- (void)connectionDidFinishLoading:(NSURLConnection *)connection

```

```

{
}

```

```

- (void)connection:(NSURLConnection *)connection didFailWithError:(NSError *)error

```

```

{

```

```

    tableView.hidden = YES;

```

```

}

```

Khi tìm kiếm thông tin có thể lựa chọn ngôn ngữ tìm kiếm, quay trở lại các thông tin được tìm kiếm trước hoặc xem tiếp các thông tin được hiển thị ở phía sau

+ Next : Chức năng xem trước.

```

- (IBAction)goBack

```

```

{

```

```

    [webView goBack];

```

```

}

```

+ Preview : Chức năng xem sau.

```

- (IBAction)goForward

```

```

{

```

```

    [webView goForward];

```

```

}

```

- Select Language : Chức năng lựa chọn ngôn ngữ

+ Sử dụng lớp thư viện : LanguageSwitcher.h

+ Code :

```

- (NSInteger)tableView:(UITableView *)tableView

```

```

numberOfRowsInSection:(NSInteger)section

```

```

{

```

```

    return [languagesArray count];
}
- (UITableViewCell *)tableView:(UITableView *)tableView
cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    static NSString *CellIdentifier = @"Cell";
    UITableViewCell *cell = [[UITableViewCell alloc]
initWithStyle:UITableViewCellStyleDefault reuseIdentifier:CellIdentifier];
    NSMutableDictionary *dictItem = [languagesArray objectAtIndex:indexPath.row];
    cell.textLabel.text = [dictItem valueForKey:@"language"];
    if ([[dictItem valueForKey:@"path"] isEqualToString:[settings
stringForKey:@"languageKey"]])
    {
        cell.accessoryType = UITableViewCellAccessoryCheckmark;
    }
    else
    {
        cell.accessoryType = UITableViewCellAccessoryNone;
    }
    return [cell autorelease];
}
- (void)tableView:(UITableView *)tableView
didSelectRowAtIndexPath:(NSIndexPath *)indexPath
{
    for (int i=0; i<[languagesArray count]; i++)
    {
        NSIndexPath *_indexPath = [NSIndexPath indexPathForRow:i
inSection:0];
        UITableViewCell *deselectedCell = [tableView
cellForRowAtIndexPath:_indexPath];
        deselectedCell.accessoryType = UITableViewCellAccessoryNone;
    }
    UITableViewCell *cell = [self.tableView cellForRowAtIndexPath:indexPath];
}

```

```

[self toggleCheckmarkedCell:cell];
[self.tableView deselectRowAtIndexPath:indexPath animated:YES];
NSMutableDictionary *dictItem = [languagesArray
objectAtIndex:indexPath.row];
[settings setObject:[dictItem valueForKey:@"path"] forKey:@"languageKey"];
[settings setObject:[dictItem valueForKey:@"language"]
forKey:@"languageName"];
}

```

- Add Bookmark : chức năng lưu trữ lại các đường dẫn đã truy cập cho lần truy cập sau.

+ Code :

```

- (IBAction)addBookmark
{
    UIAlertController *menu = [[UIAlertSheet alloc]
initWithTitle:nil
delegate:self
cancelButtonTitle:NSString(@"Cancel", @"Cancel")
destructiveButtonTitle:nil
otherButtonTitles:NSString(@"Add Bookmark", @"Add Bookmark"), nil]
}

```

- History : Chức năng lưu trữ các trang web mà người dùng đã truy cập. Người dùng có thể xóa các trang lịch sử đã được lưu lại.

+ Code :

```

- (IBAction)showHistory
{
    ModalViewController *modalView = [[ModalViewController alloc]
initWithNibName:@"ModalViewController" bundle:nil];
modalView.managedObjectContext = appDelegate.managedObjectContext;
modalView.returnValue = self;
modalView.isBookmark = NO;
[self.navigationController presentViewController:modalView
animated:YES];
}

```

```

[modalView release];
if (webView.loading)
{
    [webView stopLoading];
}
}

```

- Map : Bản đồ trực tuyến.

+ Sử dụng các hàm thư viện : UIKit.h, MapKit.h, WikiConnectionController.h, MapViewController.h.

+ Code :

- (void)locationUpdate:(CLLocation *)location

```

{
    [UIApplication sharedApplication].networkActivityIndicatorVisible = NO;
    [locationController.locationManager stopUpdatingLocation];
    [self fetchWikiPagesWithLatitude:location.coordinate.latitude
longitude:location.coordinate.longitude];
    currentLocation.latitude = location.coordinate.latitude;
    currentLocation.longitude = location.coordinate.longitude;
}

```

- (void)locationError:(NSError *)error

```

{
    UIAlertView *errorAlert = [[UIAlertView alloc]
initWithTitle:NSLocalizedString(@"Oops", @"Oops") message:@"Sorry, could not find
your location." delegate:self cancelButtonTitle:@"OK" otherButtonTitles:nil, nil];
    [errorAlert show];
    [UIApplication sharedApplication].networkActivityIndicatorVisible = NO;
    [locationController.locationManager stopUpdatingLocation];
    [errorAlert release];
}

```

3.3.Môi trường làm việc :

Để ứng dụng có thể chạy được thì điện thoại phải kết nối với Internet hoặc Wifi hoặc 3G.

Ứng dụng có thể chạy trên các phiên bản hệ điều hành iOS 3.0 trở lên.

3.4. Giao diện chương trình :





KẾT LUẬN

Qua quá trình thực hiện đề án này em đã sử dụng những kiến thức được học và cũng học hỏi thêm được rất nhiều. Em đã biết thêm được một ngôn ngữ lập trình mới. Xcode là công cụ rất hữu ích giúp cho việc lập trình được dễ dàng hơn. Với công cụ này có thể tạo ra được rất nhiều ứng dụng khác nhau cho điện thoại di động.

Ứng dụng truy cập Wikipedia trên iPhone đã phát triển được các chức năng : tìm kiếm thông tin, lựa chọn ngôn ngữ, lưu lại danh sách các đường dẫn cho lần truy cập sau, chức năng lưu trữ các trang web mà người dùng đã truy cập, bản đồ trực tuyến.

Có thể phát triển thêm các chức năng như chỉnh sửa bài viết, đăng bài viết, tải ảnh lên Wikipedia giống như truy cập Wikipedia từ máy tính.

Em sẽ cố gắng học hỏi nhiều hơn nữa để có thể tạo ra những ứng dụng khác hay hơn.

Trong quá trình làm bài em vẫn còn những sai sót mong các thầy cô giúp đỡ.

Em xin chân thành cảm ơn!