**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

**--------o0o-------**



**BÁO CÁO THỰC TẬP CƠ SỞ**

***Đề tài:***

**XÂY DỰNG CHƯƠNG TRÌNH TRÊN MÔI TRƯỜNG LINUX THỰC HIỆN QUÉT VÀ KIỂM TRA CÁC CỔNG MỞ CỦA MỘT MÁY TÍNH**

**Sinh viên thực hiện : Phạm Huy Nam**

**Lớp : ATTT-K19A**

**Mã sinh viên : DTC2054802004**

**Giáo viên hướng dẫn : Ths. Vũ Việt Dũng**

**Thái Nguyên, 03/****2023**

**MỤC LỤC**

[DANH MỤC BẢNG 5](#_Toc131506429)

[DANH MỤC HÌNH ẢNH 5](#_Toc131506430)

[LỜI MỞ ĐẦU 8](#_Toc131506431)

[Lý do chọn đề tài 8](#_Toc131506432)

[Mục đích 8](#_Toc131506433)

[Phương pháp nghiên cứu 8](#_Toc131506434)

[Kết cấu đề tài 8](#_Toc131506435)

[CHƯƠNG 1. 9](#_Toc131506436)

[TỔNG QUAN VỀ KIẾN TRÚC MÁY TÍNH 9](#_Toc131506437)

[1.1. Tìm hiểu về kiến trúc máy tính 9](#_Toc131506438)

[1.2. Kiến trúc hệ điều hành Windows, Linux, Unix 13](#_Toc131506439)

[1.2.1. Kiến trúc hệ điều hành Windows 19](#_Toc131506440)

[1.2.2. Kiến trúc hệ điều hành Linux 24](#_Toc131506441)

[1.2.3. Kiến trúc hệ điều hành Unix 28](#_Toc131506442)

[1.3. Các dịch vụ mạng phổ biến 32](#_Toc131506443)

[CHƯƠNG 2. 36](#_Toc131506444)

[TÌM HIỂU GIAO THỨC VÀ KĨ THUẬT MẠNG 36](#_Toc131506445)

[2.1. Tìm hiểu về giao thức TCP/IP 36](#_Toc131506446)

[2.2. Kĩ thuật lập trình mạng 40](#_Toc131506447)

[CHƯƠNG 3. 45](#_Toc131506448)

[ĐEMO VÀ ĐÁNH GIÁ NGUY CƠ BẢO MẬT 45](#_Toc131506449)

[3.1. Thực hiện xây dựng chương trình 45](#_Toc131506450)

[3.1.1. Tạo project trên Qt Creator 46](#_Toc131506451)

[3.1.2. Tạo giao diện chương trình Quét cổng 51](#_Toc131506452)

[3.1.3. Tạo các hàm xử lý giao diện 54](#_Toc131506453)

[3.1.4. Tạo các hàm xử lý tín hiệu 59](#_Toc131506454)

[3.1.5. Tạo hàm xuất dữ liệu và kết quả 70](#_Toc131506455)

[3.1.6. Hoàn thiện chương trình 73](#_Toc131506456)

[3.2. Đánh giá các nguy cơ đối với các cổng dịch vụ mạng 80](#_Toc131506457)

[TÀI LIỆU THAM KHẢO 86](#_Toc131506458)

# DANH MỤC BẢNG

[Bảng 1. So sánh hệ điều hành Linux và Unix 31](#_Toc131504849)

# DANH MỤC HÌNH ẢNH

[Hình 1. Một thiết kế đường ống của kiến trúc MIPS. 11](#_Toc132056855)

[Hình 2. Kiến trúc của hệ điều hành Linux 25](#_Toc132056856)

[Hình 3. Kiến trúc hệ thống Unix 27](#_Toc132056857)

[Hình 4. Các tầng trong chồng giao thức của bộ giao thức TCP/IP 36](#_Toc132056858)

[Hình 5. Hình minh hoạ giao tiếp qua các địa chỉ IP 40](#_Toc132056859)

[Hình 6. Sơ đồ cấu trúc thuật toán chương trình 44](#_Toc132056860)

[Hình 7. Giao diện Qt Creator 45](#_Toc132056861)

[Hình 8. Tạo Qt Widget Application 46](#_Toc132056862)

[Hình 9. Tạo Project Location 46](#_Toc132056863)

[Hình 10. Tạo Define Build System 47](#_Toc132056864)

[Hình 11. Tạo Class information 47](#_Toc132056865)

[Hình 12. Tạo Translation File 48](#_Toc132056866)

[Hình 13. Tạo kit selection 48](#_Toc132056867)

[Hình 14. Kết quả Project 49](#_Toc132056868)

[Hình 15. File system của chương trình 49](#_Toc132056869)

[Hình 16. Tạo file GiaoDienChuongTrinh.h 50](#_Toc132056870)

[Hình 17. Location file GiaoDienChuongTrinh.h 50](#_Toc132056871)

[Hình 18. qmake file GiaoDienChuongTrinh.h vào chương trình 51](#_Toc132056872)

[Hình 19. Tạo file GiaoDienChuongTrinh.cpp 51](#_Toc132056873)

[Hình 20. Tạo location file GiaoDienChuongTrinh.cpp 52](#_Toc132056874)

[Hình 21. Qmake file GiaoDienChuongTrinh.cpp vào chương trình 52](#_Toc132056875)

[Hình 22. Project chuongTrinhQuetCongMang 53](#_Toc132056876)

[Hình 23. Các thư viện cần thiết tạo giao diện 53](#_Toc132056877)

[Hình 24. Tạo class GiaoDien 54](#_Toc132056878)

[Hình 25. Private class GiaoDien 54](#_Toc132056879)

[Hình 26. Khởi tạo class GiaoDien 55](#_Toc132056880)

[Hình 27. Thêm các đối tượng và biến cho class GiaoDien 56](#_Toc132056881)

[Hình 28. Phần đầu của khởi tạo tab1 56](#_Toc132056882)

[Hình 29. Biến hằng của chương trình 57](#_Toc132056883)

[Hình 30. Thêm các đối tượng vào tab1 57](#_Toc132056884)

[Hình 31. Khởi tạo tab2 57](#_Toc132056885)

[Hình 32. Kết quả giao diện 58](#_Toc132056886)

[Hình 33. Các tín hiệu của chương trình 58](#_Toc132056887)

[Hình 34. Các SLOT của chương trình 59](#_Toc132056888)

[Hình 35. Các SLOT thông báo 59](#_Toc132056889)

[Hình 36. Các sự kiện tín hiệu 60](#_Toc132056890)

[Hình 37. Hàm kiểm tra IP của lineEdit 60](#_Toc132056891)

[Hình 38. Hàm thông báo lỗi 61](#_Toc132056892)

[Hình 39. Hàm kiểm tra IPv4 62](#_Toc132056893)

[Hình 40. Hàm kiểm tra trạng thái chương trình 64](#_Toc132056894)

[Hình 41. Thư viện cần cho hàm quét cổng 65](#_Toc132056895)

[Hình 42. Hàm quét cổng 66](#_Toc132056896)

[Hình 43. Hàm thông báo quét cổng 67](#_Toc132056897)

[Hình 44. Hàm Dừng chương trình 67](#_Toc132056898)

[Hình 45. Hàm thông báo Dừng chương trình 67](#_Toc132056899)

[Hình 46. Hàm reset chương trình 68](#_Toc132056900)

[Hình 47. Hàm thông báo reset 68](#_Toc132056901)

[Hình 48. Hàm thoát chương trình 68](#_Toc132056902)

[Hình 49. Các đối tượng tạo thư mục, file và hàm xử lý 69](#_Toc132056903)

[Hình 50. Hàm khởi tạo file kết quả 70](#_Toc132056904)

[Hình 51. Hàm thông báo lỗi tạo thư mục 71](#_Toc132056905)

[Hình 52. Hàm con trỏ trả về đối tượng ListWidgetItem 71](#_Toc132056906)

[Hình 53. Hàm xoá, ghi kết quả và ghi IP vào file 71](#_Toc132056907)

[Hình 54. Hàm main của chương trình 72](#_Toc132056908)

[Hình 55. kết quả của chương trình gồm các file 73](#_Toc132056909)

[Hình 56. Hộp thoại tạo file thành công 73](#_Toc132056910)

[Hình 57. Giao diện chương trình 74](#_Toc132056911)

[Hình 58. Nhập IP cần quét 74](#_Toc132056912)

[Hình 59. Thông báo quét IP hiện tại 75](#_Toc132056913)

[Hình 60. Tiến trình thay đổi 75](#_Toc132056914)

[Hình 61. Tiến trình thay đổi 76](#_Toc132056915)

[Hình 62. Thông báo hoàn thành quét cổng 76](#_Toc132056916)

[Hình 63. Kết quả các cổng mở được quét thấy 77](#_Toc132056917)

[Hình 64. Đã tạo thư mục thành công 78](#_Toc132056918)

[Hình 65. Tạo file thành công 78](#_Toc132056919)

[Hình 66. Kết quả trên file 78](#_Toc132056920)

# LỜI MỞ ĐẦU

**Lý do chọn đề tài**

Sự phát triển nhanh chóng của công nghệ thông tin và không gian mạng hiện nay đã đem lại rất nhiều lợi ích trong đời sống xã hội. Tuy nhiên, bên cạnh những lợi ích mà không gian mạng mang lại, vẫn còn nhiều nguy cơ gây ảnh hưởng đến các hệ thống thông tin và người dùng – đặc biệt là sự xuất hiện và ngày ra tăng của tội phạm không gian mạng. Trở thành mối đoẹ doạ đến bảo mật thông tin cá nhân và tổ chức doanh nghiệp. Môi trường Linux là một trong những môi trường lý tưởng mà các tội phạm mạng sử dụng để tấn công rò quét các lỗ hổng trên hệ thống. Ngoài ra, còn nhắm đến các mục tiêu phi truyền thống như các thiết bị mạng biên giới hoặc thế giới trực tuyến, số lượng, sự đa dạng và quy mô đe doạ mạng khiến đội ngũ an ninh luôn trong tình trạng cảnh giác cao độ.

Em xin trân trọng cảm ơn!

## Mục đích

Đề tài nghiên cứu nhằm mục đích:

Tìm hiểu các cổng và các dịch vụ mạng chạy trên các cổng

Xây dựng chương trình cho phép kiểm tra cổng đang mở và dịch vụ chạy trên cổng đó.

## Phương pháp nghiên cứu

Nghiên cứu lý thuyết, phân tích, đemo, kết luận.

Kết cấu đề tài

**Chương 1: Tổng quan về kiến trúc máy tính**

Tìm hiểu về kiến trúc máy tính, kiến trúc hệ điều hành Windows, Linux, Unix, các dịch vụ mạng phổ biến.

**Chương 2: Tìm hiểu giao thức và kĩ thuật mạng**

Tìm hiểu về giao thức TCP/IP, kĩ thuật lập trình mạng

**Chương 3: Demo và Đánh giá nguy cơ bảo mật**

Thực hiện xây dựng chương trình kiểm tra cổng, đánh giá nguy cơ bảo mật do dịch vụ mạng

# CHƯƠNG 1.

# TỔNG QUAN VỀ KIẾN TRÚC MÁY TÍNH

## Tìm hiểu về kiến trúc máy tính

Trong kỹ thuật máy tính, kiến trúc máy tính (tiếng Anh: computer architecture) là thiết kế khái niệm và cấu trúc hoạt động căn bản của một hệ thống máy tính. Nó là một bản thiết kế (blueprint) mô tả có tính chất chức năng về các yêu cầu (đặc biệt là tốc độ và các kết nối tương hỗ) và những sự thi hành thiết kế cho những bộ phận khác nhau của một máy tính - tập trung chủ yếu vào việc CPU hoạt động nội tại như thế nào và truy cập các địa chỉ trong bộ nhớ bằng cách nào.

Nó cũng có thể được định nghĩa như là khoa học và nghệ thuật lựa chọn và kết nối các thành phần phần cứng để tạo thành các máy tính đáp ứng được các mục đích về tính năng, hiệu suất và giá cả.

Kiến trúc máy tính bao gồm ít nhất ba phạm trù con chính:

***Một là***, Kiến trúc tập lệnh (Instruction set architecture, ISA), là hình ảnh trừu tượng của một hệ thống tính toán được nhìn từ góc độ của một lập trình viên sử dụng ngôn ngữ máy (hay hợp ngữ), bao gồm tập lệnh, cách đánh địa chỉ bộ nhớ (memory address modes), các thanh ghi, và các định dạng địa chỉ và dữ liệu.

Kiến trúc tập lệnh đóng vai trò rất quan trọng trong việc thiết kế, hiện thực bộ xử lý bên dưới và thiết kế, hiện thực các phần mềm hoạt động trên nền bộ xử lý. Có hai hướng thiết kế kiến trúc tập lệnh là kiến trúc CISC (Complex Instruction Set Computer) và RISC (Reduced Instruction Set Computer).

Cùng với sự phát triển của phần cứng máy tính, một trong những sự phát triển khác dễ dàng được nhận thấy là các ngôn ngữ lập trình. Ngày càng có nhiều các ngôn ngữ lập trình mạnh và phức tạp được phát triển. Các ngôn ngữ lập trình này giúp cho các lập trình viên phát triển ứng dụng dễ dàng hơn, chính xác hơn và có thể đi sâu vào chi tiết hơn. Tuy nhiên, điều này làm nảy sinh một khoảng cách ngữ nghĩa (semantic gap) giữa những tác vụ được cung cấp bởi ngôn ngữ lập trình cấp cao và những tác vụ được cung cấp bởi kiến trúc của một máy tính. Biểu hiện rõ nhất của khoảng cách này là sự thực thi không hiệu quả của các ngôn ngữ cấp cao, kích thước chương trình tăng, độ phức tạp của các trình biên dịch cũng tăng theo. Do vậy, nhiều kỹ sư thiết kế phần cứng có khuynh hướng giải quyết vấn đề này bằng cách xây dựng những kiến trúc máy tính hỗ trợ một số lượng lớn các loại lệnh phức tạp với nhiều phương pháp tính toán địa chỉ khác nhau, và nhiều lệnh của ngôn ngữ cấp cao sẽ được hiện thực hoàn toàn bằng phần cứng. Mục tiêu chính của hướng tiếp cận này là:

* Làm giảm độ phức tạp của trình biên dịch.
* Cải tiến thời gian thực thi vì các tác vụ phức tạp có thể được hiện thực bằng phần cứng.
* Hoàn thành tác vụ với ít số lượng lệnh hợp ngữ.

Đây chính là hướng tiếp cận theo kiến trúc CISC. Một kiến trúc tập lệnh kiểu CISC thông thường có khoảng từ 120 đến 350 lệnh sử dụng nhiều loại định dạng lệnh và dữ liệu khác nhau. Tuy nhiên, số lượng các thanh ghi trong các kiến trúc CISC thường khá ít khi chỉ có khoảng từ 8 đến 24 thanh ghi đa dụng (general-purpose registers - GPRs). Các kiến trúc CISC thực hiện phần lớn các tác vụ tham khảo bộ nhớ thông qua rất nhiều phương pháp tính toán địa chỉ khác nhau.

Tuy vậy, các kiến trúc CISC thường không đạt hiệu suất sử dụng phần cứng cao cũng như các mục tiêu của kiến trúc CISC nêu trên thường không đạt được. Các nhà khoa học máy tính nhận thấy rằng, chỉ khoảng 25% các lệnh phức tạp được sử dụng ở mức 95% thời gian; tức là có đến 75% phần cứng hỗ trợ các lệnh phức tạp không thường được sử dụng. Do đó, tài nguyên phần cứng thường bị lãng phí trong kiến trúc CISC. Mục tiêu giảm độ phức tạp của các trình biên dịch thường cũng khó đạt được do các trình biên dịch thường khó khăn trong việc tìm kiếm các lệnh phức tạp trong một ứng dụng nào đó sao cho hoàn toàn phù hợp với kiến trúc phần cứng đã định sẵn. Ngoài ra, việc định thời và đặc biệt là tổ chức thực thi các lệnh theo cơ chế xử lý ống (pipeline) là khá khó khăn cho các lệnh phức tạp. Ngày nay, khi mà giá thành các bộ nhớ ngày càng giảm thì mục tiêu giảm số lượng lệnh hợp ngữ cũng không phải là quá quan trọng.

Các kiến trúc sư phần cứng máy tính do đó chuyển sang hướng tiếp cận theo kiến trúc RISC. Mặc dù được định nghĩa và được thiết kế khác nhau giữa các nhóm nghiên cứu cũng như các công ty sản xuất bộ xử lý, đặc điểm chung và nổi bật của các bộ xử lý được thiết kế theo kiến trúc RISC là:

* Bao gồm nhiều thanh ghi đa dụng.
* Tập lệnh đơn giản và giới hạn.
* Tập trung vào cơ chế xử lý ống.

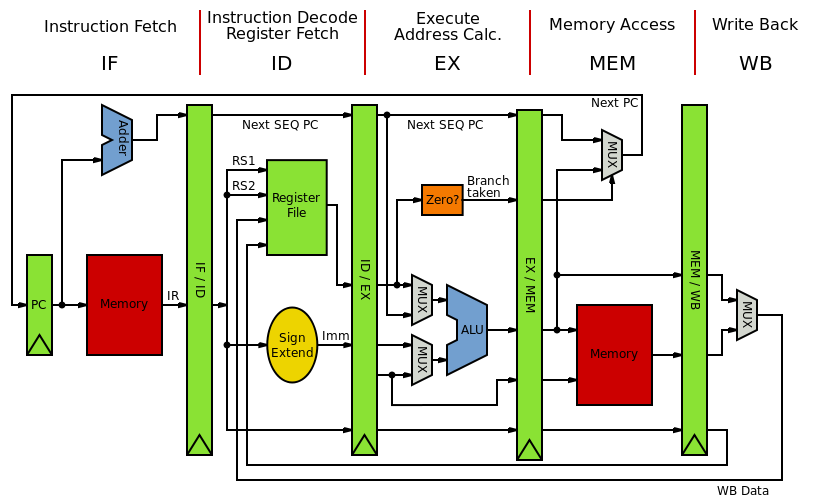
Một bộ xử lý được thiết kế theo kiến trúc RISC thường bao gồm ít hơn 100 lệnh với kích thước cố định (thường là 32 bit). Các lệnh thường hoạt động dựa trên các thanh ghi trong khi bộ nhớ được truy xuất chỉ thông qua các lệnh đọc/ghi (load/store). Hầu hết các lệnh trong kiến trúc RISC được hoàn thành trong một chu kỳ xung nhịp, do đó việc xử lý theo cơ chế ống là khả thi. Các đặc tính này dẫn đến kết quả là bộ xử lý có thể hoạt động ở tần số xung nhịp cao hơn và CPI (cycles per instruction) thấp hơn, do đó hiệu suất đạt được sẽ cao hơn. Ngoài ra, tài nguyên phần cứng cần thiết cho bộ xử lý kiến trúc RISC cũng thường thấp hơn so với kiến trúc CISC do phần cứng chỉ hỗ trợ những lệnh đơn giản.

***Hai là***, Vi kiến trúc (Microarchitecture), còn gọi là Tổ chức máy tính (Computer organization) là một mô tả bậc thấp, cụ thể hơn về hệ thống. Mô tả này nói về các bộ phận cấu thành của hệ thống được kết nối với nhau như thế nào và chúng hoạt động tương hỗ như thể nào để thực hiện kiến trúc tập lệnh. Ví dụ, kích thước bộ đệm cache của một máy tính là một đặc điểm về tổ chức máy tính mà thường không liên quan đến kiến trúc tập lệnh [1].

Thiết kế vi kiến trúc là quá trình xây dựng nên một bộ xử lý cụ thể dựa trên một kiến trúc tập lệnh sao cho các lệnh trong kiến trúc tập lệnh có thể được xử lý tự động bởi bộ xử lý. Thiết kế vi kiến trúc một bộ xử lý là một quá trình phức tạp đòi hỏi nhiều công sức. Mục tiêu của thiết kế vi kiến trúc không những phải bảo đảm rằng tất cả các lệnh đã được thiết kế trong kiến trúc tập lệnh phải được thực thi chính xác mà còn là việc tối ưu sự thực thi sao cho đạt được hiệu suất cao nhất, ít tốn tài nguyên phần cứng và năng lượng tiêu thụ ít nhất.

Việc thực thi các lệnh trong các bộ xử lý thường trải qua năm bước là: đọc lệnh (instruction fetch – IF), giải mã lệnh (instruction decode – ID), thực thi (execute – EX), truy xuất bộ nhớ dữ liệu (memory access – MEM) và cập nhật kết quả (write back – WB). Quá trình thiết kế vi kiến trúc của bộ xử lý thường cũng sẽ đi theo các giai đoạn này. Ở mỗi giai đoạn sẽ cần những khối chức năng (functional units) khác nhau như bộ nhớ chương trình, tập thanh ghi, khối luận lý số học,… Các khối này phải được lựa chọn (chức năng, kích thước các ngõ vào ra,…) và kết nối với nhau một cách khoa học và nghệ thuật nhằm đảm bảo các yêu cầu vừa nêu.

Hình 1. Một thiết kế đường ống của kiến trúc MIPS.



***Ba là***, Thiết kế hệ thống (System Design) bao gồm tất cả các thành phần phần cứng khác bên trong một hệ thống tính toán chẳng hạn:

* các đường kết nối hệ thống như bus (máy tính) và switch
* các bộ điều khiển bộ nhớ (memory controller) và các cây phả hệ bộ nhớ
* các cơ chế CPU off-load như Direct memory access (truy nhập bộ nhớ trực tiếp)
* các vấn đề như đa xử lý (multi-processing).

Thiết kế hệ thống máy tính là công việc xây dựng một hệ thống máy chính hoàn chỉnh bao gồm thiết kế các bộ nhớ phân cấp, thiết kế các thiết bị ngoại vi… Trong đó, thiết kế bộ nhớ phân cấp chủ yếu tập trung vào tối ưu hoá việc tổ chức bộ nhớ đệm theo các kỹ thuật khác nhau như: ánh xạ trực tiếp (direct mapped), ánh xạ kết hợp (n-way set associative) và ánh xạ toàn phần (fully associative). Bộ nhớ đệm trong một hệ thống bộ nhớ phân cấp của một hệ thống máy tính có vai trò rất quan trọng trong hoạt động của bộ xử lý nói riêng và của cả hệ thống máy tính nói chung. Tổ chức bộ nhớ phân cấp ảnh hưởng trực tiếp đến hiệu suất của máy tính bởi vì mỗi truy xuất ở cấp bộ nhớ thấp hơn sẽ tốn rất nhiều thời gian (có thể gấp hang trăm chu kỳ so với truy xuất ở cấp cao hơn). Việc lựa chọn các thiết bị ngoại vi cũng sẽ ảnh hưởng lớn đến hiệu suất của cả hệ thống máy tính vì thời gian đáp ứng ngoại vi ảnh hưởng đến thời gian xử lý ứng dụng của bộ xử lý [2].

## Kiến trúc hệ điều hành Windows, Linux, Unix

Hệ điều hành cung cấp môi trường cho các chương trình thực thi. Các hệ điều hành rất khác biệt nhau về kiến trúc, chúng được tổ chức cùng với các dòng khác nhau. Thiết kế một hệ điều hành mới là một công việc quan trọng. Các mục đích của hệ thống phải được định nghĩa rõ ràng trước khi thiết kế bắt đầu. Kiểu hệ thống mong muốn là cơ sở cho việc chọn lựa giữa các giải thuật và chiến lược khác nhau.

Các thành phần của hệ thống bao gồm:

***Một là***, Quản lý tiến trình của hệ thống. Một tiến trình là một chương trình đang được thi hành, nhưng ý nghĩa của nó còn rộng hơn. Một công việc theo lô là một tiến trình. Một chương trình người dùng chia sẻ thời gian là một tiến trình, một công việc của hệ thống như soopling xuất ra máy in cũng là một tiến trình.

Một tiến trình phải sử dụng tài nguyên như thời gian sử dụng CPU, bộ nhớ, tập tin, các thiết bị nhập xuất để hoàn tất công việc của nó. Các tài nguyên này được cung cấp khi tiến trình được tạo hay trong quá trình thi hành. Khi tiến trình được tạo, nó sử dụng rất nhiều tài nguyên vật lý và luận lý. Cũng như một số khởi tạo dữ liệu nhập. Ví dụ , khảo sát tiến trình hiển thị trạng thái của tập tin lên màn hình. Đầu vào của tiến trình là tên tập tin, và tiến trình sẽ thực hiện những chỉ thị thích hợp, thực hiện lời gọi hệ thống để nhận được những thông tin mong muốn và hiển thị nó lên màn hình. Khi tiến trình kết thúc, hệ điều hành sẽ tái tạo lại các tài nguyên có thể được dùng lại..

Một tiến trình là hoạt động (active) hoàn toàn-ngược lại với một tập tin trên đĩa là thụ động (passive)-với một bộ đếm chương trình cho biết lệnh kế tiếp được thi hành.Việc thi hành được thực hiện theo cơ chế tuần tự , CPU sẽ thi hành từ lệnh đầu đến lệnh cuối.

Một tiến trình được coi là một đơn vị làm việc của hệ thống. Một hệ thống có thể có nhiều tiến trình cùng lúc, trong đó một số tiến trình là của hệ điều hành, một số tiến trình là của người sử dụng. các tiến trình này có thể diễn ra đồng thời.

Vai trò của hệ điều hành trong việc quản lý tiến trình là :

* Tạo và hủy các tiến trình của người sử dụng và của hệ thống.
* Ngưng và thực hiện lại một tiến trình.
* Cung cấp cơ chế đồng bộ tiến trình.
* Cung cấp cách thông tin giữa các tiến trình.
* Cung cấp cơ chế kiểm soát deadlock

***Hai là***, Quản lý bộ nhớ chính. Trong hệ thống máy tính hiện đại, bộ nhớ chính là trung tâm của các thao tác, xử lý. Bộ nhớ chính có thể xem như một mảng kiểu byte hay kiểu word. Mỗi phần tử đều có địa chỉ. Đó là nơi lưu dữ liệu được CPU truy xuất một cách nhanh chóng so với các thiết bị nhập/xuất. CPU đọc những chỉ thị từ bộ nhớ chính. Các thiết bị nhập/xuất cài đặt cơ chế DMA(xem chương IV) cũng đọc và ghi dữ liệu trong bộ nhớ chính. Thông thường bộ nhớ chính chứa các thiết bị mà CPU có thể định vị trực tiếp.

Một chương trình muốn thi hành trước hết phải được ánh xạ thành địa chỉ tuyệt đối và nạp vào bộ nhớ chính.Khi chương trình thi hành, hệ thống truy xuất các chỉ thị và dữ liệu của chương trình trong bộ nhớ chính. Ngay cả khi tiến trình kết thúc , dữ liệu vẫn còn trong bộ nhớ cho đến khi một tiến trình khác được ghi chồng lên.

Hệ điều hành có những vai trò trong việc quản lý bộ nhớ chính :

* Lưu giữ thông tin về các vị trí trong bộ nhớ đã được sử dụng và ai sử dụng.
* Quyết định tiến trình nào được nạp vào bộ nhớ chính, khi bộ nhớ đã có thể dùng được.
* Cấp phát và thu hồi bộ nhớ khi cần thiết.

***Ba là***, Quản lý bộ nhớ phụ. Những chương trình với dữ liệu truy xuất của chúng phải được đặt trong bộ nhớ chính trong suốt quá trình thi hành. Nhưng bộ nhớ chính quá nhỏ để có thể lưu giữ mọi dữ liệu và chương trình, ngoài ra dữ liệu sẽ mất khi không còn được cung cấp năng lượng. Hệ thống máy tính ngày nay cung cấp hệ thống lưu trữ phụ. Đa số các máy tính đều dùng đĩa để lưu trữ cả chương trình và dữ liệu. Hầu như tất cả chương trình: chương trình dịch, hợp ngữ, thủ tục, trình soạn thảo, định dạng... đều được lưu trữ trên đĩa cho tới khi nó được thực hiện, nạp vào trong bộ nhớ chính và cũng sử dụng đĩa để chứa dữ liệu và kết quả xử lý. Vì vậy một bộ quản lý hệ thống đĩa rất quan trọng cho hệ thống máy tính.

Vai trò của hệ điều hành trong việc quản lý đĩa :

* Quản lý vùng trống trên đĩa.
* Định vị lưu trữ.
* Lập lịch cho đĩa.
* Vì hệ thống đĩa được sử dụng thường xuyên, nên nó phải được dùng hiệu quả.Tốc độ của toàn bộ hệ thống tuỳ thuộc rất nhiều vào tốc độ truy xuất đĩa.

***Bốn là***, Quản lý hệ thống nhập xuất. Một trong những mục tiêu của hệ điều hành là che dấu những đặc thù của các thiết bị phần cứng đối với người sử dụng thay vào đó là một lớp thân thiện hơn, người sử dụng dể thao tác hơn.

Một hệ thống nhập/xuất bao gồm :

* Hệ thống buffer caching.
* Giao tiếp điều khiển thiết bị (device drivers) tổng quát.
* Bộ điều khiển cho các thiết bị phần cứng.

Chỉ có device driver mới hiểu đến cấu trúc đặc thù của thiết bị mà nó mô tả.

***Năm là***, Quản lý hệ thống tập tin. Quản lý tập tin là một trong những thành phần có thể nhìn thấy nhất của hệ điều hành. Máy tính có thể lưu thông tin trên nhiều loại phương tiện lưu trữ vật lý khác nhau. Băng từ, đĩa từ, đĩa quang là những phương tiện thông dụng nhất. Mỗi phương tiện này có đặc điểm và tổ chức riêng. Mỗi phương tiện được điều khiển bởi một thiết bị, như một ổ đĩa hay ổ băng từ. Các thuộc tính này bao gồm tốc độ truy xuất, dung lượng, tốc độ truyền dữ liệu và phương pháp truy xuất.

Tập tin là tập hợp thông tin có quan hệ được định nghĩa bởi người tạo. Thông thường, các tập tin biểu diễn chương trình và dữ liệu. Các tập tin dữ liệu có thể là số, chữ cái, chữ số. Các tập tin có dạng bất kỳ (thí dụ, các tập tin văn bản) hay có thể được định dạng có cấu trúc (thí dụ, các trường cố định). Một tập tin chứa một chuỗi các bits, bytes, các dòng hay các mẫu tin mà ý nghĩa của nó được định nghĩa bởi người tạo. Khái niệm tập tin là một khái niệm cực kỳ thông dụng.

Hệ điều hành có nhiệm vụ thực hiện các hoạt động trong việc quản lý hệ thống tập tin:

* Tạo và xoá tập tin
* Tạo và xoá thư mục
* Hỗ trợ các hàm nguyên thuỷ để thao tác tập tin và thư mục
* Ánh xạ các tập tin trên các thiết bị lưu trữ phụ
* Sao lưu dự phòng tập tin trên các phương tiện lưu trữ ổ định

***Sáu là***, Hệ thống bảo vệ. Trong một hệ thống nhiều người sử dụng và cho phép nhiều tiến trình diễn ra đồng thời, các tiến trình phải được bảo vệ đối với những hoạt động khác.Do đó, hệ thống cung cấp cơ chế để đảm bảo rằng tập tin, bộ nhớ, CPU, và những tài nguyên khác chỉ được truy xuất bởi những tiến trình có quyền.

Hệ thống bảo vệ là một cơ chế kiểm soát quá trình truy xuất của chương trình, tiến trình, hoặc người sử dụng với tài nguyên của hệ thống. Cơ chế này cũng cung cấp cách thức để mô tả lại mức độ kiểm soát.

***Bảy là***, hệ thống cơ chế dòng lệnh. Một trong những phần quan trọng của chương trình hệ thống trong một hệ điều hành là cơ chế dòng lệnh, đó là giao tiếp giữa người sử dụng và hệ điều hành. Một số hệ điều hành đặt cơ chế dòng lệnh bên trong hạt nhân, số khác như MS-DOS và UNIX thì xem hệ điều hành như là một chương trình đặt biệt, được thi hành khi các công việc bắt đầu hoặc khi người sử dụng login lần đầu tiên.

Các lệnh đưa vào hệ điều hành thông qua bộ điều khiển lệnh. Trong các hệ thống chia xẻ thời gian một chương trình có thể đọc và thông dịch các lệnh điều khiển được thực hiện một cách tự động. Chương trình này thường được gọi là bộ thông dịch điều khiển card, cơ chế dòng lệnh hoặc Shell. Chức năng của nó rất đơn giản đó là lấy lệnh kế tiếp và thi hành.

Mỗi hệ điều hành sẽ có những giao tiếp khác nhau, dạng đơn giản theo cơ chế dòng lệnh, dạng thân thiện với người sử dụng như giao diện của Macintosh có các biểu tượng, cửa sổ thao tác dùng chuột.

Các lệnh có quan hệ với việc tạo và quản lý các tiến trình, kiểm soát nhập xuất, quản lý bộ lưu trữ phụ, quản lý bộ nhớ chính, truy xuất hệ thống tập tin và cơ chế bảo vệ.

***Tám là***, Mạng hệ thống. Hệ phân tán là tập hợp các bộ xử lý, chúng không chia sẻ bộ nhớ, các thiết bị ngoại vi hay đồng hồ. Thay vào đó mỗi bộ xử lý có bộ nhớ, đồng hồ và các bộ xử lý giao tiếp với nhau thông qua các đường giao tiếp như bus tốc độ cao hay mạng. Các bộ xử lý trong hệ thống phân tán khác nhau về kích thước và chức năng. Chúng có thể chứa các bộ vi xử lý, trạm làm việc, máy vi tính và các hệ thống máy tính thông thường.

Các bộ xử lý trong hệ thống được nối với nhau thông qua mạng truyền thông có thể được cấu hình trong nhiều cách khác nhau. Mạng có thể được nối kết một phần hay toàn bộ. Thiết kế mạng truyền thông phải xem xét vạch đường thông điệp và các chiến lược nối kết, và các vấn đề cạnh tranh hay bảo mật.

Hệ điều hành thường tổng quát hoá việc truy xuất mạng như một dạng truy xuất tập tin, với những chi tiết mạng được chứa trong trình điều khiển thiết bị của giao diện mạng. Các giao thức tạo một hệ thống phân tán có thể có một ảnh hưởng to lớn trên tiện ích và tính phổ biến của hệ thống đó. Sự đổi mới của World Wide Web đã tạo ra một phương pháp truy xuất mới cho thông tin chia sẻ. Nó đã cải tiến giao thức truyền tập tin (File Transfer Protocol-FTP) và hệ thống tập tin mạng (Network File System-NFS) đã có bằng cách xoá yêu cầu cho một người dùng đăng nhập trước khi người dùng đó được phép dùng tài nguyên ở xa. Định nghĩa một giao thức mới, giao thức truyền siêu văn bản (hypertext transfer protocol-http), dùng trong giao tiếp giữa một trình phục vụ web và trình duyệt web. Trình duyệt web chỉ cần gởi yêu cầu thông tin tới một trình phục vụ web của máy ở xa, thông tin (văn bản, đồ hoạ, liên kết tới những thông tin khác) được trả về.

***Chín là***, Các dịch vụ hệ điều hành, Hệ điều hành cung cấp các dịch vụ xác định tới chương trình và tới người dùng của các chương trình đó. Dĩ nhiên, các dịch vụ được cung cấp khác nhau từ hệ điều hành này với hệ điều hành kia nhưng chúng có thể xác định các lớp chung. Các dịch vụ hệ điều hành được cung cấp sự tiện dụng cho người lập trình để thực hiện tác vụ lập trình dễ dàng.

Các dịch vụ của hệ điều hành gồm:

*Thực thi chương trình:* hệ thống phải có thể nạp chương trình vào bộ nhớ và chạy chương trình đó. Chương trình phải có thể kết thúc việc thực thi của nó bình thường hay không bình thường (hiển thị lỗi).

*Thao tác xuất/nhập:* một chương trình đang chạy có thể yêu cầu xuất/nhập. Xuất/nhập này có thể liên quan tới tập tin hay thiết bị xuất/nhập. Đối với các thiết bị cụ thể, các chức năng đặc biệt có thể được mong muốn (như quay lại từ đầu một ổ băng từ, hay xoá màn hình). Đối với tính hiệu quả và tính bảo vệ, người dùng thường không thể điều khiển các thiết bị xuất/nhập trực tiếp. Do đó, hệ điều hành phải cung cấp một phương tiện để thực hiện xuất/nhập..

*Thao tác hệ thống tập tin:* hệ thống tập tin có sự quan tâm đặc biệt. Các chương trình cần đọc từ và viết tới các tập tin. Chương trình cũng cần tạo và xoá tập tin bằng tên.

*Giao tiếp:* trong nhiều trường hợp, một quá trình cần trao đổi thông tin với các quá trình khác. Giao tiếp như thế có thể xảy ra trong hai cách chính. Cách đầu tiên xảy ra giữa các quá trình được thực thi trên cùng máy tính; cách thứ hai xảy ra giữa hai quá trình đang được thực thi trên các máy tính khác nhau được kết nối với nhau bởi một mạng máy tính. Các giao tiếp có thể được thực hiện bằng bộ nhớ được chia sẻ, hay bằng kỹ thuật truyền thông điệp, trong đó các gói tin được di chuyển giữa các quá trình bởi hệ điều hành.

*Phát hiện lỗi:* hệ điều hành liên tục yêu cầu nhận biết các lỗi có thể phát sinh. Các lỗi có thể xảy ra trong CPU và phần cứng bộ nhớ (như lỗi bộ nhớ hay lỗi về điện), trong các thiết bị xuất/nhập (như lỗi chẳn lẻ trên băng từ, lỗi nối kết mạng, hết giấy in) và trong chương trình người dùng (như tràn số học, cố gắng truy xuất một vị trí bộ nhớ không hợp lệ, dùng quá nhiều thời gian CPU). Đối với mỗi loại lỗi, hệ điều hành nên thực hiện một hoạt động hợp lý để đảm bảo tính toán đúng và không đổi.

***Mười là***, Lời gọi hệ thống. Lời gọi hệ thống cung cấp một giao tiếp giữa tiến trình và hệ điều hành. Lời gọi này cũng như các lệnh hợp ngữ.

Một số hệ thống cho phép lời gọi hệ thống được thực hiện từ cấp lập trình ngôn ngữ cấp cao, như các hàm và lời gọi hàm. Nó có thể phát sinh lời gọi từ các thủ tục hay gọi trực tiếp trong dòng.

Trong các ngôn ngữ lập trình cấp cao, người sử dụng không cần quan tâm đến chi tiết mà chỉ cần thông qua các hàm hay các lệnh để thực hiện.Lời gọi hệ thống có thể diễn ra theo một cách khác. Kiểu và khối lượng thông tin tùy thuộc vào hệ thống và lúc gọi.

Có ba phương pháp được sử dụng để chuyển tham số cho hệ điều hành. Cách đơn giản nhất là chuyển tham số vào thanh ghi. Nếu có nhiều tham số, nó sẽ được lưu trữ trong khối hoặc bảng trong bộ nhớ. Cách cuối cùng là dùng cơ chế stack.

Lời gọi hệ thống có thể được chia thành các loại : kiểm soát tiến trình, thao tác tập tin, thao tác thiết bị, thông tin [3].

### Kiến trúc hệ điều hành Windows

Microsoft Windows (hoặc đơn giản là Windows) là tên của một hệ điều hành dựa trên giao diện người dùng đồ hoạ được phát triển và được phân phối bởi Microsoft. Nó bao gồm một vài các dòng hệ điều hành, mỗi trong số đó phục vụ một phần nhất định của ngành công nghiệp máy tính.

Phiên bản đầu tiên của hệ điều hành Windows được Microsoft ra mắt vào năm 1985, là hệ điều hành đầu tiên có giao diện đồ hoạ của hãng này, với tên gọi Windows 1.0 – tên mã nội bộ là Interface Manager.

Tên gọi Windows được lựa chọn bởi hệ điều hành của Microsoft xoay quanh những khung nội dung hình chữ nhật hiển thị trên màn hình. Trải qua nhiều phiên bản với nhiều sự thay đổi, đến nay hệ điều hành Windows đã rất thành công trong việc chiếm lĩnh thị trường.

Trải qua nhiều phiên bản với nhiều sự thay đổi, hệ điều hành Windows đã gặt hái rất nhiều thành công trong đó phải kể đến Windows XP, Windows 7 và Window 8.

Ngoài ra, các phiên bản khác của Windows như: Window 98, Windows 2000, Windows Vista, Windows Sever… và gần đây nhất là Windows 10, Windows 11.

Vốn là một nền tảng chiếm thị phần sử dụng cao nhất hiện nay nên cũng không có gì khó hiểu khi hầu hết các nhà sản xuất đều đầu tư xây dựng phần mềm cũng như sản xuất phần cứng hỗ trợ cho hệ điều hành Windows.

Có thể nói rằng, Windows hỗ trợ đầy đủ các ứng dụng bạn cần và nhiều hơn rất nhiều, vượt trội hơn so với các hệ điều hành khác. Đơn giản một điều, các nhà viết ứng dụng luôn muốn nhắm đến một thị trường có số người sử dụng đông đảo như Windows.

Song song với những thành ưu điểm thì Windows vẫn còn những khuyết điểm hiện hữu.

Hiện nay, tỷ lệ người sử dụng Windows không bản quyền đang rất cao mà đặc biệt là Việt Nam. Có nhiều nguyên nhân khách quan dẫn đến vấn đề này nhưng nếu bạn là người nguyên tắc và tôn trọng bản quyền thì nên sử dụng các dòng máy được hỗ trợ sẵn hệ điều hành bản quyền.

Là đối tượng mục tiêu của nhiều tin tặc – hacker. Do được sử dụng quá phổ biến nên nền tảng Windows tập trung rất nhiều sự chú ý của Hacker, vì thế phần lớn các virus, phần mềm gián điệp, mã độc… đều được viết để hoạt động trên nền tảng này. Việc phòng chống virus luôn phải được cập nhật thường xuyên và đầy đủ.

Cấu tạo chính của Hệ điều hành Windows bao gồm:

***Cấu hình và bảo trì – Configuration and maintenance***: được thực hiện thông qua nhiều công cụ và tính năng khác nhau:

* **Control Panel:** cho phép quản lý các cài đặt hệ thống như màn hình, âm thanh, thiết bị, tài khoản người dùng, v.v.
* **Group Policy Editor:** cho phép quản lý các cài đặt hệ thống trên một mạng nội bộ.
* **Registry Editor:** cho phép quản lý các cài đặt hệ thống chi tiết hơn bằng cách chỉnh sửa registry.
* **Windows Update:** cập nhật các bản vá bảo mật và các tính năng mới cho hệ thống.
* **Disk Cleanup:** giúp xóa các tệp không cần thiết và tăng không gian đĩa cứng.
* **Windows Defender:** cung cấp bảo vệ chống lại phần mềm độc hại và các cuộc tấn công từ bên ngoài.
* **Task Manager:** giúp quản lý các tiến trình đang chạy trên hệ thống và đóng các tiến trình không cần thiết.
* **Device Manager:** cho phép quản lý các thiết bị phần cứng, bao gồm cài đặt và cập nhật trình điều khiển.
* **Performance Monitor**: cung cấp thông tin về tình trạng hoạt động của các phần cứng và phần mềm trên hệ thống.

***Giao diện người dùng – User interface*** là cách mà người dùng tương tác với hệ thống.

* **Desktop:** Desktop là giao diện người dùng chính trên Windows. Nó bao gồm một tấm nền, thanh taskbar và các biểu tượng ứng dụng.
* **Start Menu:** Start Menu là điểm khởi đầu của hệ thống Windows. Nó cung cấp truy cập đến các ứng dụng, tập tin và cài đặt hệ thống.
* **File Explorer:** File Explorer là giao diện người dùng để quản lý tập tin trên hệ thống. Nó cho phép người dùng duyệt qua các thư mục và tập tin trên hệ thống, sao chép, di chuyển, xóa và tạo mới các tập tin.

***Ứng dụng và tiện ích – Applications and utilities:*** Một số ứng dụng và tiện ích phổ biến trên Windows:

* **Microsoft Office:** Microsoft Office bao gồm các ứng dụng như Word, Excel, PowerPoint và Outlook. Đây là một bộ công cụ văn phòng cực kỳ hữu ích cho việc xử lý văn bản, tính toán, thiết kế bài thuyết trình và quản lý email.
* **Google Chrome:** Đây là trình duyệt web phổ biến nhất trên Windows, cung cấp cho người dùng trải nghiệm duyệt web nhanh và tiện lợi.
* **VLC Media Player:** Đây là một trình phát đa phương tiện mạnh mẽ, hỗ trợ nhiều định dạng tệp âm thanh và video.

***Các thành phần của Windows Server – Windows Server components:*** Nó cung cấp nhiều thành phần khác nhau để hỗ trợ các chức năng khác nhau của máy chủ. Dưới đây là một số thành phần quan trọng của Windows Server:

* **Active Directory:** Đây là một dịch vụ thư mục của Windows Server, cung cấp cho người dùng và các ứng dụng khác khả năng xác thực và quản lý tài khoản người dùng.
* **Domain Name System (DNS):** DNS là một dịch vụ quan trọng của Windows Server, cho phép các máy tính kết nối với nhau thông qua tên miền thay vì địa chỉ IP.
* **Dynamic Host Configuration Protocol (DHCP):** DHCP cung cấp các địa chỉ IP động cho các máy tính trong một mạng, giúp đơn giản hóa việc quản lý các địa chỉ IP trong một mạng lớn.

***Hệ thống tập tin – File systems***: hệ thống tập tin phổ biến nhất trên Windows:

* **NTFS (New Technology File System):** NTFS là hệ thống tập tin mặc định được sử dụng trên Windows 10/8/7/Vista/XP. NTFS hỗ trợ bảo mật cao, phân quyền tệp tin và thư mục, đường dẫn tên tệp lên đến 255 ký tự, tối đa dung lượng phân vùng 16 EB.
* **FAT32 (File Allocation Table):** FAT32 là một hệ thống tập tin cũ hơn, được sử dụng trên các phiên bản cũ hơn của Windows, như Windows 95, 98 và ME. FAT32 hỗ trợ dung lượng tối đa cho mỗi tệp là 4GB và dung lượng phân vùng tối đa là 2TB.
* **exFAT (Extended File Allocation Table):** exFAT là một hệ thống tập tin mới hơn, được sử dụng để hỗ trợ việc lưu trữ dữ liệu trên các thiết bị lưu trữ di động như USB, thẻ nhớ và ổ cứng ngoài. exFAT hỗ trợ dung lượng phân vùng lớn hơn 32GB và tối đa dung lượng tệp tin là 16EB.

***Các thành phần cốt lõi (core components)***trên hệ điều hành Windowsbao gồm:

* **Windows Kernel:** Là phần cốt lõi của hệ điều hành, quản lý tài nguyên phần cứng và cho phép các ứng dụng tương tác với phần cứng. Kernel cung cấp các tính năng như quản lý bộ nhớ, lập lịch và xử lý lỗi.
* **Windows Shell:** Là môi trường đồ họa của Windows, cung cấp giao diện người dùng để tương tác với hệ thống. Shell cung cấp các tính năng như quản lý tệp tin, các ứng dụng và các thiết bị ngoại vi.
* **Device Drivers:** Là các chương trình được thiết kế để cho phép hệ thống tương tác với phần cứng, bao gồm các driver cho các thiết bị như bàn phím, chuột, đầu đọc thẻ, màn hình, âm thanh, mạng,...

***Dịch vụ (Services) trên hệ điều hành Windows*** là các chương trình chạy ngầm mà không có giao diện người dùng trực tiếp. Các dịch vụ này được thiết kế để thực hiện các tác vụ chạy nền hoặc tự động, phục vụ cho việc quản lý và điều khiển hệ thống, các ứng dụng và các thiết bị:

* **Print Spooler:** Dịch vụ này quản lý và xử lý các tác vụ in ấn trên hệ thống.
* **Remote Registry:** Dịch vụ này cho phép truy cập và thay đổi các cài đặt và thông tin trong Registry từ xa.
* **Windows Update:** Dịch vụ này cung cấp các bản cập nhật bảo mật, các bản vá lỗi và các bản cập nhật phần mềm khác cho hệ thống.
* **Windows Firewall:** Dịch vụ này cung cấp tính năng tường lửa, giúp bảo vệ hệ thống khỏi các cuộc tấn công mạng.

***DirectX*** là một bộ các thư viện phần mềm của Microsoft được sử dụng để hỗ trợ việc xử lý đồ họa, âm thanh, mạng và các tính năng khác trên hệ thống Windows. DirectX được phát triển để giúp các nhà phát triển tạo ra các ứng dụng đa phương tiện trên nền tảng Windows một cách dễ dàng và hiệu quả hơn.

***Mạng – Networking:*** hỗ trợ việc kết nối và quản lý mạng trên các thiết bị:

* **Network Adapter:** Đây là thiết bị phần cứng trên máy tính để kết nối với mạng. Windows hỗ trợ nhiều loại card mạng, bao gồm Ethernet, Wi-Fi, Bluetooth và các thiết bị khác.
* **TCP/IP:** TCP/IP là giao thức mạng được sử dụng rộng rãi trên Internet và các mạng nội bộ. Windows hỗ trợ các phiên bản của giao thức này để kết nối và truyền dữ liệu trên mạng.
* **Network Sharing:** Windows cung cấp chức năng chia sẻ tài nguyên như thư mục, máy in, kết nối Internet và các thiết bị khác giữa các máy tính trong mạng.

***Kịch bản và dòng lệnh (scripting and command-line)*** là một phương pháp sử dụng các lệnh và kịch bản để thực hiện các tác vụ trên hệ thống máy tính của bạn. Trong Windows, bạn có thể sử dụng dòng lệnh và kịch bản để thực hiện các tác vụ như tạo, sao chép và xóa tệp, quản lý quyền truy cập, cài đặt phần mềm, và nhiều hơn nữa.

***Hạt nhân (Kernel) trên Windows*** là một phần của hệ điều hành Windows và là trung tâm của hệ thống. Nó là một chương trình có quyền cao nhất trong hệ thống và quản lý tất cả các tài nguyên và dịch vụ của hệ thống, bao gồm bộ nhớ, thiết bị đầu cuối, trình điều khiển, quản lý quyền truy cập và nhiều hơn nữa. Hạt nhân Windows NT được thiết kế để hỗ trợ các kiến trúc máy tính khác nhau, bao gồm x86, x64, ARM và IA-64. Nó cung cấp một số chức năng cốt lõi như quản lý bộ nhớ, bảo mật, hệ thống tập tin, và các tính năng mạng.

***.NET Framework*** là một nền tảng phần mềm được phát triển bởi Microsoft, cung cấp một môi trường thực thi để chạy các ứng dụng trên hệ điều hành Windows. .NET Framework cung cấp các công cụ và thư viện để phát triển các ứng dụng Windows, web và di động.

***APIs (Application Programming Interfaces) trên Windows*** là một tập hợp các hàm và thủ tục có thể được gọi bởi một ứng dụng phần mềm để tương tác với hệ điều hành. Các APIs này cung cấp quyền truy cập đến các tài nguyên hệ thống khác nhau, chẳng hạn như tập tin, bộ nhớ, mạng và các thiết bị phần cứng, cho phép ứng dụng thực hiện các nhiệm vụ như đọc và ghi dữ liệu, hiển thị giao diện người dùng đồ họa hoặc giao tiếp qua mạng. Có một số loại APIs khác nhau trên Windows, bao gồm:

* **Win32 API:** Đây là API phổ biến nhất được sử dụng cho các ứng dụng desktop Windows. Nó cung cấp quyền truy cập đến một loạt các dịch vụ hệ thống, chẳng hạn như các điều khiển giao diện người dùng, tập tin và mạng I/O, và cấu hình hệ thống.
* **.NET Framework API:** Đây là một tập hợp các APIs cung cấp một framework để phát triển các ứng dụng Windows sử dụng nền tảng .NET. Nó bao gồm một số lượng lớn các lớp và thư viện được xây dựng sẵn cho các nhiệm vụ thông thường như truy cập cơ sở dữ liệu, xử lý XML và các dịch vụ web.

***Bảo vệ (Security) trên Windows*** là một chủ đề quan trọng để bảo vệ hệ thống khỏi các mối đe dọa bảo mật. Hệ điều hành Windows cung cấp nhiều tính năng bảo vệ để giúp đảm bảo rằng dữ liệu và hệ thống của người dùng được bảo mật và an toàn. Một số tính năng bảo vệ chính trên Windows bao gồm:

***Firewall:*** Windows có một tường lửa tích hợp để giúp ngăn chặn các kết nối không mong muốn đến hệ thống của bạn từ Internet hoặc mạng cục bộ.

***Windows Defender:*** Đây là một ứng dụng chống virus, phần mềm gián điệp và phần mềm độc hại tích hợp trên Windows. Nó cung cấp khả năng quét máy tính để phát hiện và loại bỏ các phần mềm độc hại, bảo vệ người dùng khỏi các cuộc tấn công mạng và cung cấp tính năng tường lửa để ngăn chặn các cuộc tấn công từ bên ngoài.

***User Account Control (UAC):*** Đây là một tính năng bảo mật trên Windows giúp ngăn chặn các chương trình độc hại khỏi việc thực hiện các thay đổi trên hệ thống của người dùng bằng cách yêu cầu xác nhận từ người dùng trước khi thực hiện các hoạt động quan trọng như cài đặt phần mềm mới hay thay đổi các cài đặt hệ thống [4].

### Kiến trúc hệ điều hành Linux

Lịch sử của Linux bắt đầu vào năm 1991 với sự bắt đầu của một dự án cá nhân của sinh viên Phần Lan Linus Torvalds để tạo ra một hạt nhân hệ điều hành tự do mới. Kể từ đó, các kết quả của nhân Linux đã được tăng trưởng liên tục trong suốt lịch sử của nó. Kể từ khi phát hành mã nguồn của nó lần đầu vào năm 1991, nó đã phát triển từ một số nhỏ các tập tin viết bằng C theo một giấy phép cấm phân phối thương mại đến các phiên bản 3.10 vào năm 2013 với hơn 16 triệu dòng mã nguồn, và đến bản phát hành 4.15 năm 2008 nó đã lên có 23.3 triệu dòng lệnh. dưới Giấy phép Công cộng GNU [5].

Linux là một hệ điều hành nhân Unix, linh động và được phát hành miễn phí. Cùng với Windows, Mac OS thì Linux là một trong những hệ diều hành phổ biến nhất hiện nay [6].

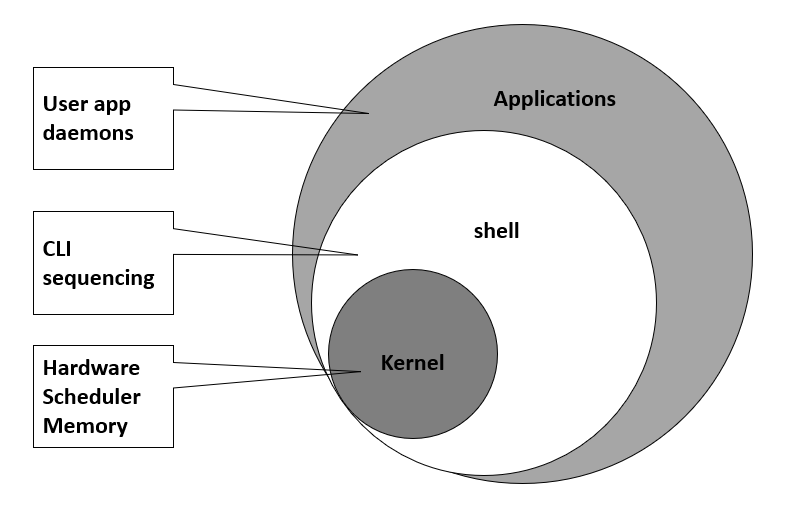
Linux là một hệ điều hành phát triển dựa vào hệ điều hành Unix và được phát hành miễn phí. Hệ điều hành này được cài đặt từ máy tính cá nhân đến các server chuyên dụng.

Thành phần cơ bản của Linux là hạt nhân Linux (Linux kernel), là nhân hệ điều hành được phát triển bởi Linus Torvalds. Linux được công bố lần đầu tiên vào tháng 9 năm 1991 với phiên bản 0.01. Linux hiện có khá nhiều các bản phân phối khác nhau tất cả đều có một đặc điểm chung là có một Linux kernel bên trong.

Kiến trúc của hệ điều hành Linux chia làm 3 thành phần chính: Kernel, Shell, Applications.

* **Kernel:** Đây là phần quan trọng và được ví như trái tim của hệ điều hành, phần kernel chứa các module, thư viện để quản lý và giao tiếp với phần cứng và các ứng dụng.
* **Shell:** Shell là một chương trình có chức năng thực thi các lệnh từ người dùng hoặc từ các ứng dụng – tiện ích yêu cầu chuyển đến cho Kernel xử lý.
* **Applications:** Là các ứng dụng và tiện ích mà người dùng cài đặt trên Server. Ví dụ: ftp, samba, Proxy, …

Ngoài ra, hệ thống Linux còn có một số thành phần khác như các tệp hệ thống, giao diện người dùng, các tiến trình, quản lý gói phần mềm, hệ thống tập tin, hệ thống mạng và bảo mật. Tất cả các thành phần này được tổ chức và tương tác với nhau để tạo thành một hệ thống hoạt động mạnh mẽ, ổn định và bảo mật.



Hình 2. Kiến trúc của hệ điều hành Linux

Ưu điểm của hệ điều hành Linux:

* **Miễn phí và mã nguồn mở:** Linux là miễn phí và có mã nguồn mở, có nghĩa là bất kỳ ai đều có thể sử dụng, phân phối, tùy chỉnh, sửa đổi và phát triển các phiên bản mới của nó.
* **Bảo mật:** Linux có kiến ​​trúc bảo mật cao và ít bị tấn công bởi virus, malware và phần mềm độc hại. Nó được thiết kế để chống lại các cuộc tấn công mạng và bảo vệ dữ liệu của người dùng.
* **Ổn định:** Linux là hệ điều hành ổn định và có thể chạy một cách liên tục trong nhiều tháng hoặc năm mà không cần phải khởi động lại.
* **Đa dạng:** Linux có thể được cài đặt trên nhiều loại thiết bị, từ máy tính để bàn đến máy chủ, thiết bị nhúng và thiết bị di động.
* **Tiết kiệm tài nguyên:** Linux sử dụng ít tài nguyên hệ thống hơn so với các hệ điều hành khác, cho phép nó chạy trên các thiết bị với phần cứng yếu hơn.
* **Quản lý gói:** Linux có hệ thống quản lý gói tiện lợi, giúp người dùng cài đặt và cập nhật phần mềm một cách dễ dàng.

Nhược điểm của hệ điều hành Linux:

* **Khó sử dụng cho người mới:** Linux có một số khó khăn khi sử dụng cho người mới bắt đầu, nhất là nếu họ không có kinh nghiệm với các hệ điều hành dòng lệnh hoặc mã nguồn mở.
* **Hỗ trợ phần cứng hạn chế:** Linux chưa được hỗ trợ phần cứng đầy đủ bởi các nhà sản xuất phần cứng, làm cho việc sử dụng Linux trên một số thiết bị trở nên khó khăn.
* **Khó khăn trong việc cài đặt phần mềm:** Một số phần mềm phổ biến không có phiên bản cho Linux hoặc cài đặt phần mềm trên Linux có thể phức tạp hơn so với các hệ điều hành khác.
* **Thiếu ứng dụng:** Một số ứng dụng văn phòng, trò chơi và phần mềm đồ họa chuyên nghiệp không có.

Các distro phổ biến của Linux:

***Debian*** là một distro chứa số lượng các phần mềm rất lớn. Debian được xây dựng bởi một tổ chức nguyên tình nguyện cống hiến để phát triển phần mềm tự do và đẩy mạnh những lý tưởng của cộng đồng phần mềm tự do. Debian có tiếng về mối liên kết chặt chẽ với triết lí Unix và phần mềm tự do. Nó cũng có tiếng về sự phong phú cho các chọn lựa: Phiên bản phát hành hiện tại có hơn 29000 gói phần mềm cho 11 kiến trúc máy tính, từ kiến trúc ARM thường gặp ở các hệ thống nhúng và kiến trúc máy tính lớn s390 của IBM cho đến các kiến trúc thường gặp trên máy tính cá nhân hiện đại như x86 và Power.

***Hệ điều hành Ubuntu*** là một trong những bản phân phối Linux phổ biến nhất hiện nay do Mark Shuttleworth sáng lập và công ty Canonical của ông tài trợ. Hệ điều hành này được sử dụng phổ biến và ưa chuộng vì có giao diện đẹp, thân thiện, dễ sử dụng, kho phần mềm ứng dụng rất phong phú đáp ứng được hầu hết yêu cầu của người dùng, các version được cập nhật liên tục. Ubuntu được chia làm 2 loại: Ubuntu Desktop và Ubuntu Server.

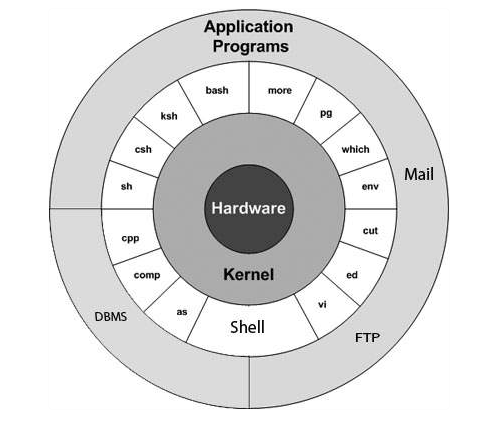
***Kali Linux*** là một distro của Linux, được sử dụng để kiểm tra, tấn công thử nghiệm vào các lỗ hổng của hệ thống công nghệ thông tin. Kali linux được viết dựa trên nền tảng của Debian được đồng bộ hóa với các Repository của Debian. Đây là một phiên bản tiến hóa của Backtrack và distro này rất hữu ích đối với những chuyên gia đánh giá bảo mật.

### Kiến trúc hệ điều hành Unix

Unix (được đăng ký nhãn hiệu là UNIX) là một họ hệ điều hành máy tính đa nhiệm, đa người dùng được viết vào những năm 1960 và 1970 do một số nhân viên của Bell Labs thuộc AT&T bao gồm Ken Thompson, Dennis Ritchie và Douglas McIlroy và một số người khác [7].

Hệ thống Unix bao gồm một số thành phần ban đầu được đóng gói cùng nhau. Bằng cách bao gồm môi trường phát triển, thư viện, tài liệu và mã nguồn di động, có thể sửa đổi cho tất cả các thành phần này, ngoài hạt nhân của hệ điều hành, Unix là một hệ thống phần mềm độc lập. Đây là một trong những lý do chính khiến nó nổi lên như một công cụ dạy và học quan trọng và có ảnh hưởng rộng rãi như vậy.

Unix được thiết kế dành riêng cho các lập trình viên. Ban đầu, hệ điều hành này dự định sẽ được sử dụng bên trong hệ thống Bell. Tuy nhiên, vào cuối những năm 1970, AT&T đã cấp phép UNIX cho các tổ chức bên ngoài sử dụng. Nơi nó trở thành hệ điều hành được sử dụng rộng rãi cho các máy chủ internet, máy trạm và hệ thống máy tính lớn.



Hình 3. Kiến trúc hệ thống Unix

Hệ điều hành Unix gồm có 4 thành phần cơ bản:

**Kernel:** Bộ phận này thực hiện các chức năng hệ điều hành cơ bản. Bao gồm truy cập tệp, xử lý thông tin liên lạc và cấp phát bộ nhớ. Nó cũng bao gồm cấu hình, trình điều khiển thiết bị, cấu trúc tệp, quản lý bộ nhớ, cuộc gọi hệ thống, v.v.

**Shell:** Một chương trình mở rộng chạy liên tục để cung cấp giao diện tương tác giữa người dùng, mã nguồn và các chức năng máy tính. Shell sử dụng cú pháp chuẩn cho tất cả các lệnh. C Shell, Bourne Shell và Korn Shell là những shell được biết đến nhiều nhất và có sẵn trong hầu hết các phiên bản Unix.

**Document (tài liệu):** Tài liệu trong hệ điều hành Unix bao gồm: các trang thủ công và các tệp lớn hơn trình bày chi tiết các hệ thống con chính. Tất cả các file được tổ chức vào trong các thư mục. Những thư mục này được tổ chức trong một cấu trúc dạng cây được gọi như là hệ thống file.

**Các lệnh:** Các lệnh Unix rất đa dạng và có phân biệt chữ hoa, chữ thường. Có trên 250 lệnh tiêu chuẩn cộng với một số lệnh khác được cung cấp bởi phần mềm thứ 3. Tất cả các lệnh này đi cùng với các tùy chọn (chức năng) của nó. Tổ hợp các lệnh trong Unix cho phép người dùng điều hướng hệ điều hành và thực hiện các hành động cụ thể, cũng như các ứng dụng tiện ích chung và bảo trì.

Hệ điều hành Unix đang được sử dụng rất rộng rãi trên toàn cầu. Mặc dù vậy, sự xuất hiện của hệ điều hành này tại Việt Nam vẫn chưa thực sự được ưa chuộng tại Việt Nam.

Ưu điểm của hệ điều hành Unix:

* Unix là hệ điều hành đa nhiệm, đa người dùng. Nó chứa các tài nguyên thông tin giống nhau và có thể dễ dàng chia sẻ bởi bất cứ người dùng nào. Nên rất thuận tiện trong công việc.
* Hệ điều hành này cung ứng nhiều tác vụ và cho phép người dùng thực hiện cùng lúc nhiều quy trình khác nhau.
* Unix là hệ điều hành đầu tiên trên thế giới sử dụng ngôn ngữ C cao cấp. Nó giúp cho bạn có thể chuyển sang các máy tính có sự thích ứng thấp nhất.
* Khả năng tích hợp tốt giúp người dùng dễ dàng trao đổi thông tin trên các ứng dụng chạy trên hệ điều hành.
* Đồng thời, Unix cũng cung cấp cấu trúc tệp phân cấp, giúp cho việc truy cập và bảo trì dữ liệu thuận tiện hơn.
* Unix có thể mở rộng các chức năng thông qua các chương trình xây dựng trên giao diện lập trình tiêu chuẩn.

Bên cạnh rất nhiều ưu điểm tuyệt vời, hệ điều hành Unix vẫn có những hạn chế, nhược điểm mà các lập trình cần quan trọng trong quá trình ứng dụng vào công việc, sáng tạo. Dưới đây là một vài nhược điểm của Unix.

Unix có rất nhiều phiên bản khác nhau. Nhưng giữa chúng lại có rất nhiều khác biệt, thiếu đi tính đồng bộ. Vì thế, bạn sẽ gặp khó khăn trong việc ứng dụng phiên bản mới. Bởi phiên bản kế nhiệm có nhiều thay đổi nên các lập trình sẽ phải trau dồi thêm kiến thức.

Vì Unix là hệ điều hành đa nhiệm nên khi sử dụng, bạn phải đảm bảo máy tính có bộ xử lý mạnh, bộ nhớ trong lớn cùng nhiều thiết bị ngoại vi khác nhau.

Hệ điều hành Unix có rất nhiều phiên bản. Bản mới nhất là Unix V7. Trước đó, 2 phiên bản chính của hệ điều hành này, gồm: System V Release 4 do AT&T cấp chứng nhận và một bản khác là BSD 4.4 từ đại học California (Berkeley). Trong đó, System V Release 4 là phiên bản thành công nhất của Unix về thương mại.

Hiện nay, có nhiều hệ điều hành được kế thừa và phát triển từ Unix. Đó là: Mac OS X, Linux, Android, iOS, Chrome OS, Orbis OS. Chúng được sử dụng trên PlayStation 4. Tất cả các hệ điều hành này thường được gọi chung là hệ điều hành “Unix-like”.

|  |  |  |
| --- | --- | --- |
| Hạng mục | Linux | Unix |
| Mô hình nguồn | Sử dụng mã nguồn mở | Phần lớn là là độc quyền (một số ít bản phân phối là mã nguồn mở) |
| Giấy phép hoạt động | Giấy phép công cộng (cho phép người dùng sử dụng tự do) | Giấy phép độc quyền (cần đăng ký, mua) |
| Chi phí đầu tư | Thường miễn phí (một số bản phân phối cao cấp hơn sẽ phải trả phí, nhưng không quá lớn) | Chi phí đắt hơn (Unix cũng có một số bản phân phối miễn phí hoặc phụ thuộc vào thiết bị mà bạn sử dụng) |
| Ngôn ngữ | Sử dụng đa ngôn ngữ | Ngôn ngữ chính là tiếng Anh |
| Tiêu chuẩn | Không có tiêu chuẩn hóa hiện đại | Đặc điểm UNIX duy nhất |
| Phương pháp tiếp cận hạt nhân | Phương pháp tiếp cận nguyên khối | Nguyên khối, microkernel hoặc hybrid |
| Lớp mặc định bên ngoài | BASH (Bourne Again Shell) | Bourne Shell |
| Tốc độ phân giải và nhận dạng mối đe dọa | Nhanh hơn (dựa trên cộng đồng) | Chậm hơn (phụ thuộc vào bản vá) |
| Đối tượng sử dụng | Các chuyên gia, các tổ chức hoặc cá nhân | Chuyên gia, các nhà phát triển chuyên biệt |
| Các ví dụ về phân phối | Ubuntu, Fedora, Redhat, Android | Solaris, HP-UX, macOS |

Bảng 1. So sánh hệ điều hành Linux và Unix

## Các dịch vụ mạng phổ biến

Internet hay Mạng là một hệ thống thông tin toàn cầu có thể được truy nhập công cộng gồm các mạng máy tính được liên kết với nhau. Hệ thống này truyền thông tin theo kiểu nối chuyển gói dữ liệu (packet switching) dựa trên một giao thức liên mạng đã được chuẩn hóa (giao thức IP). Hệ thống này bao gồm hàng ngàn mạng máy tính nhỏ hơn của các doanh nghiệp, của các viện nghiên cứu và các trường đại học, của người dùng cá nhân và các chính phủ trên toàn cầu, được liên kết bởi một loạt các công nghệ mạng điện tử, không dây và mạng quang. Internet mang theo một loạt các tài nguyên và dịch vụ thông tin, chẳng hạn như các tài liệu và ứng dụng siêu văn bản được liên kết với nhau của World Wide Web (WWW), thư điện tử, điện thoại và chia sẻ file.

Internet không có tổ chức quản trị tập trung duy nhất nào trong việc thực hiện công nghệ hoặc chính sách cho truy cập và sử dụng; mỗi mạng cấu thành đặt chính sách riêng của mình. Các định nghĩa của hai không gian tên chính trong Internet, không gian địa chỉ Giao thức Internet (địa chỉ IP) và Hệ thống tên miền (DNS), được chỉ đạo bởi một tổ chức bảo trì, Tập đoàn Internet về Tên miền và số được gán (ICANN). Nền tảng kỹ thuật và tiêu chuẩn hóa các giao thức cốt lõi là một hoạt động của Lực lượng đặc nhiệm kỹ thuật Internet (IETF), một tổ chức phi lợi nhuận của những người tham gia quốc tế liên kết lỏng lẻo mà bất kỳ ai cũng có thể liên kết bằng cách đóng góp chuyên môn kỹ thuật. Vào tháng 11 năm 2006 [8], Internet đã được đưa vào danh sách Bảy kỳ quan mới của USA Today.

Với sự phát triển mạnh mẽ như hiện nay, mạng internet đã trở thành một cơ sở công cộng phép hàng trăm triệu người trên thế giới cùng một lúc truy cập. Khi sử dụng web bạn có thể truy cập vào rất rất nhiều thông tin khác nhau. Tùy thuộc vào trình duyệt bạn sử dụng, giao diện xuất hiện website cũng có sự khác biệt.

***Một số công cụ tìm kiến trên Internet:***

**GOOGLE:** Nói đến google chắc chúng ta không còn xa lạ nữa, vì sự phổ biến của nó trên khắp thế giới. Đây là dịch vụ nắm khoảng 65% thị trường Mỹ. Google có các tính năng hỗ trợ người dùng tìm kiếm những thông tin có trên Internet. Các trang web của Google (ví dụ: Google Search, Gmail, Google Drive) thường hoạt động trên các cổng mạng khác nhau, tùy thuộc vào các giao thức và phương thức truyền tải dữ liệu. Ví dụ, trang web Google Search sử dụng giao thức HTTP (HyperText Transfer Protocol) và hoạt động trên cổng mạng 80 hoặc 8080, trong khi trang web Gmail sử dụng giao thức HTTPS (HTTP Secure) và hoạt động trên cổng mạng 443.

**BING:** Giống như google, Bing cũng là công cụ tìm kiếm phổ biến tại Mỹ. Nó sẽ hỗ trợ tìm được thông tin trả lời bạn cần nhanh chóng, quyết định nhiều thông tin khác nhau. Tại Mỹ nó là dịch vụ Internet đứng thứ 2 với khoảng 10%. Như Google, khi truy cập vào trang web của Bing hoặc sử dụng các dịch vụ khác của Microsoft như Outlook hoặc OneDrive, các trang web này sẽ sử dụng các cổng khác nhau, phụ thuộc vào giao thức và phương thức truyền tải dữ liệu. Ví dụ, trang web Bing sử dụng giao thức HTTPS và hoạt động trên cổng mạng 443.

**WOLFRAM ALPHA:** Đây là công cụ tính toán vào kiến thức là chính, đáp ứng nhu cầu thông tin định lượng. Nếu bạn đang cần tìm thông tin có tính thống kê hoặc thực tế, mà thời điểm đó bạn đang ở khu vực nhất. WOLFRAM ALPHA sẽ truy vấn dữ liệu khổng lồ của mình, mang tới thông tin giúp bạn trả lời trực tiếp, mà không cần nó tới trang web ngoài.

**YAHOO:** Yahoo là công cụ tìm kiếm rất nổi tiếng trên thế giới, ngoài ra web Yahoo còn là 1 trong những trang truy cập nhiều nhất trên mạng. Yahoo đáp ứng các thông tin như Hỏi đáp, trò chuyện, tìm kiếm, hình ảnh, … Nó được biết là cổng web của mình, Yahoo Directory, Yahoo Mail, Yahoo News, quảng cáo, bản đồ trực tuyến, Video, và những phương tiện truyền thông, dịch vụ khác.

**BAIDU:** Baidu là công cụ tìm kiếm lớn nhất Trung Quốc, tương tự như Google của Mỹ. Baidu có đủ những thông tin về tin tức, hình ảnh.. khi truy cập vào trang web của Baidu hoặc sử dụng các dịch vụ khác của Baidu như Baidu Maps hoặc Baidu Tieba, các trang web này sẽ sử dụng các cổng khác nhau, phụ thuộc vào giao thức và phương thức truyền tải dữ liệu. Ví dụ, trang web Baidu sử dụng giao thức HTTPS và có thể hoạt động trên cổng mạng 443 [9].

***Một số dịch vụ mạng phổ biến:***

**World Wide Web (WWW):** Là một hệ thống thông tin toàn cầu được truy cập thông qua Internet. Người dùng có thể truy cập các trang web, blog và các trang thông tin trên các chủ đề khác nhau. World Wide Web (WWW) hoạt động trên cổng mặc định là cổng 80. Tuy nhiên, khi sử dụng kết nối bảo mật HTTPS thì trang web hoạt động trên cổng 443.

**Email:** Là dịch vụ truyền thông điện tử giữa các người dùng trên mạng. Người dùng có thể gửi và nhận thư điện tử (email) thông qua Internet. Email hoạt động trên nhiều cổng khác nhau, tùy thuộc vào giao thức mà nó sử dụng. Các cổng phổ biến nhất được sử dụng cho email là:

* **SMTP (Simple Mail Transfer Protocol)** sử dụng cổng 25 hoặc cổng 587
* **POP3 (Post Office Protocol version 3)** sử dụng cổng 110
* **IMAP (Internet Message Access Protocol)** sử dụng cổng 143 hoặc cổng 993 (đối với IMAP over SSL/TLS)

Tuy nhiên, nếu sử dụng các giao thức bảo mật như SMTP over SSL/TLS hoặc IMAP over SSL/TLS thì các cổng sử dụng sẽ khác nhau. Ví dụ, SMTP over SSL/TLS sử dụng cổng 465.

**File Transfer Protocol (FTP):** Là dịch vụ cho phép người dùng truyền tải tập tin giữa các máy tính trên mạng. File Transfer Protocol (FTP) sử dụng hai cổng khác nhau để hoạt động, đó là cổng 20 và cổng 21. Cổng 21 được sử dụng cho kết nối điều khiển và cổng 20 được sử dụng cho truyền dữ liệu. Tuy nhiên, khi sử dụng giao thức FTP qua SSL/TLS (FTPS), các cổng sử dụng sẽ khác nhau tùy thuộc vào cách cấu hình của máy chủ FTP. Thông thường, các cổng sử dụng cho FTPS là cổng 990 hoặc các cổng ngẫu nhiên được chọn trong phạm vi 1024 đến 65535.

**Instant Messaging (IM):** Là dịch vụ trò chuyện trực tiếp giữa các người dùng trên mạng. Người dùng có thể trao đổi tin nhắn văn bản, âm thanh và hình ảnh thông qua các ứng dụng như Skype, WhatsApp, Facebook Messenger, v.v. Instant Messaging (IM) thường sử dụng các giao thức khác nhau, và do đó có thể hoạt động trên nhiều cổng khác nhau tùy thuộc vào ứng dụng cụ thể. Tuy nhiên, các ứng dụng phổ biến như Skype, WhatsApp và Facebook Messenger thường sử dụng các cổng như sau:

* **Skype:** sử dụng các cổng 80, 443, 3478, 3479 và 3480
* **WhatsApp:** sử dụng các cổng 80, 443, 5222, 5223 và 5228
* **Facebook Messenger:** sử dụng các cổng 80, 443 và 5222.

Lưu ý rằng các ứng dụng này có thể sử dụng các cổng khác nhau tùy thuộc vào cách cấu hình và phiên bản của chúng.

**Social Media:** Là các dịch vụ trực tuyến cho phép người dùng tạo hồ sơ cá nhân và chia sẻ thông tin với những người khác trên mạng. Các mạng xã hội phổ biến như Facebook, Twitter, Instagram và LinkedIn. Social Media thường hoạt động thông qua các ứng dụng web và di động, và sử dụng các giao thức khác nhau để truyền tải dữ liệu. Do đó, các mạng xã hội không hoạt động trên một cổng cụ thể. Thay vào đó, chúng sử dụng các cổng phổ biến như cổng 80 (HTTP) và cổng 443 (HTTPS) để truyền tải dữ liệu giữa máy chủ và trình duyệt của người dùng. Ngoài ra, các ứng dụng di động cũng có thể sử dụng các cổng khác nhau để truyền tải dữ liệu qua các kết nối mạng di động hoặc Wi-Fi.

**Voice over Internet Protocol (VoIP):** Là dịch vụ cho phép người dùng thực hiện các cuộc gọi điện thoại qua Internet, thay vì sử dụng hệ thống điện thoại truyền thống. Voice over Internet Protocol (VoIP) thường sử dụng các giao thức khác nhau để truyền tải âm thanh qua mạng Internet, nhưng phổ biến nhất là giao thức SIP (Session Initiation Protocol). Giao thức này thường sử dụng cổng 5060 cho các yêu cầu kết nối và cổng 5061 cho các kết nối bảo mật sử dụng giao thức TLS (Transport Layer Security). Ngoài ra, các dịch vụ VoIP như Skype, Google Voice và WhatsApp có thể sử dụng các cổng khác nhau tùy thuộc vào cách cấu hình và phiên bản của chúng.

**Virtual Private Network (VPN):** Là dịch vụ cho phép người dùng truy cập vào mạng Internet công cộng một cách an toàn và riêng tư. VPN cho phép người dùng kết nối đến các máy tính và mạng riêng tư khác nhau trên toàn thế giới. Virtual Private Network (VPN) sử dụng các giao thức khác nhau để truyền tải dữ liệu qua mạng Internet, nhưng phổ biến nhất là giao thức OpenVPN. Giao thức này sử dụng cổng 1194 để truyền tải dữ liệu qua kết nối VPN. Ngoài ra, các giao thức khác như PPTP (Point-to-Point Tunneling Protocol) và L2TP (Layer 2 Tunneling Protocol) cũng có thể sử dụng các cổng khác nhau để truyền tải dữ liệu qua kết nối VPN.

# CHƯƠNG 2.

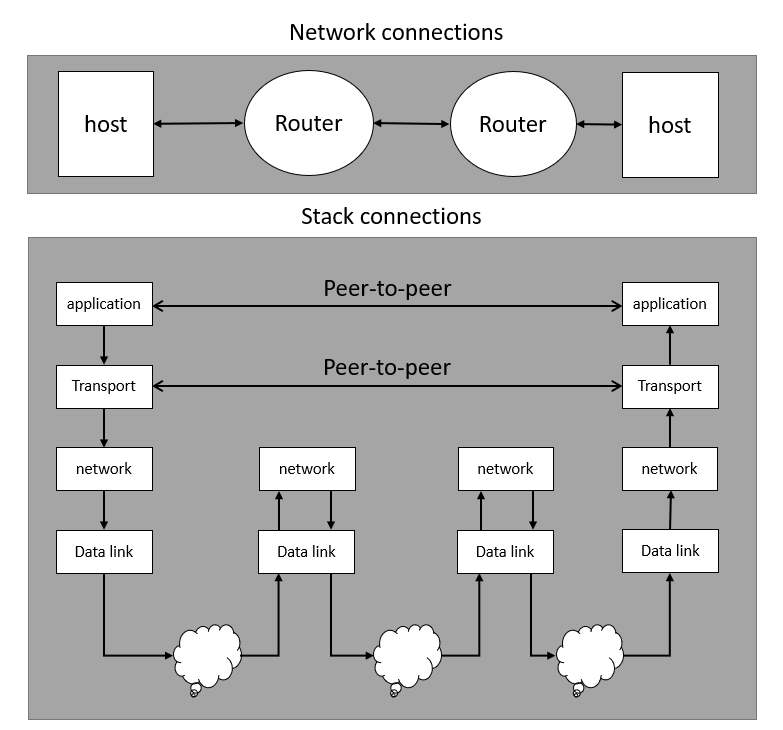
# TÌM HIỂU GIAO THỨC VÀ KĨ THUẬT MẠNG

## Tìm hiểu về giao thức TCP/IP

TCP/IP là cụm từ viết tắt của Transmission Control Protocol/Internet Protocol hay còn gọi là giao thức điều khiển truyền nhận/ Giao thức liên mạng. Đây là một bộ các giao thức truyền thông được sử dụng để kết nối các thiết bị mạng với nhau trên internet. TCP/IP cũng có thể được sử dụng như một giao thức truyền thông trong mạng máy tính riêng (mạng nội bộ). Trong đó, bộ Giao thức internet – một tập hợp các quy tắc và thủ tục – thường gọi là TCP/IP (TCP/IP Protocol) TCP và IP là hai giao thức chính bên cạnh những giao thức khác trong bộ. Bộ giao thức TCP/IP hoạt động như một lớp trừu tượng giữa các ứng dụng internet và hạ tầng router/switch.TCP/IP chỉ định cách dữ liệu được trao đổi qua internet. Nó thực hiện bằng cách cung cấp thông tin liên lạc đầu cuối. Từ đó xác định cách nó được chia thành các packet, xác định địa chỉ, truyền dẫn, định tuyến và nhận dữ liệu. TCP/IP được thiết kế để đảm bảo độ tin cậy, nó có khả năng khôi phục tự động khi gặp sự cố trong quá trình truyền dữ liệu.

Bộ giao thức liên mạng xuất phát từ công trình DARPA, từ những năm đầu thập niên kỷ 1970. Sau khi đã hoàn thành việc xây dựng ARPANET tiên phong, DARPA bắt đầu công việc trên một số những kỹ thuật truyền thông dữ liệu khác. Vào năm 1972, Robert E. Kahn đã được thuê vào làm việc tại Văn phòng kỹ thuật điều hành tin tức (Information Processing Technology Office) của DARPA, phòng có chức năng liên quan đến mạng lưới truyền thông dữ liệu thông qua vệ tinh và mạng lưới truyền thông bằng sóng radio trên mặt đất. Trong quá trình làm việc tại đây Kahn đã phát hiện ra giá trị của việc liên thông giữa chúng [10].

Có bốn phiên bản đã được xây dựng - TCP v1, TCP v2. Phiên bản 3 được tách ra thành hai phần TCP v3 và IP v3, vào mùa xuân năm 1978, và sau đó ổn định hóa với phiên bản TCP/IP v4 - giao thức tiêu chuẩn hiện dùng của Internet ngày nay.



Hình 4. Các tầng trong chồng giao thức của bộ giao thức TCP/IP

***Tầng ứng dụng(Application):*** DNS, TFTP, TLS/SSL, FTP, HTTP, IMAP, IRC, NNTP, POP3, SIP, SMTP, SNMP, SSH, TELNET, ECHO, BitTorrent, RTP, PNRP, rlogin, ENRP, …Các giao thức định tuyến như BGP và RIP, vì một số lý do, chạy trên TCP và UDP - theo thứ tự từng cặp: BGP dùng TCP, RIP dùng UDP - còn có thể được coi là một phần của tầng ứng dụng hoặc tầng mạng.

Tầng ứng dụng là nơi các chương trình mạng thường dùng làm việc nhất nhằm liên lạc giữa các nút trong một mạng. Do chồng TCP/IP không có tầng nào nằm giữa ứng dụng và các tầng giao vận, tầng ứng dụng trong bộ TCP/IP phải bao gồm các giao thức hoạt động như các giao thức tại tầng trình diễn và tầng phiên của mô hình OSI. Việc này thường được thực hiện qua các thư viện lập trình.

Hai giao thức tầng thấp thông dụng nhất là TCP và UDP. Mỗi ứng dụng sử dụng dịch vụ của một trong hai giao thức trên đều cần có cổng. Hầu hết các ứng dụng thông dụng có các cổng đặc biệt được cấp sẵn cho các chương trình phục vụ (server)(HTTP - Giao thức truyền siêu văn bản dùng cổng 80; FTP - Giao thức truyền tệp dùng cổng 21, v.v..) trong khi các trình khách (client) sử dụng các cổng tạm thời (ephemeral port).

***Tầng giao vận(Transport):*** TCP, UDP, DCCP, SCTP, IL, RUDP, …Các giao thức định tuyến như OSPF (tuyến ngắn nhất được chọn đầu tiên), chạy trên IP, cũng có thể được coi là một phần của tầng giao vận, hoặc tầng mạng. ICMP (Internet control message protocol là Giao thức điều khiển thông điệp Internet) và IGMP (Internet group management protocol - tạm dịch là Giao thức quản lý nhóm Internet) chạy trên IP, có thể được coi là một phần của tầng mạng.

Trách nhiệm của tầng giao vận là kết hợp các khả năng truyền thông điệp trực tiếp (end-to-end) không phụ thuộc vào mạng bên dưới, kèm theo kiểm soát lỗi (error control), phân mảnh (fragmentation) và điều khiển lưu lượng. Việc truyền thông điệp trực tiếp hay kết nối các ứng dụng tại tầng giao vận có thể được phân loại như sau:

* định hướng kết nối (connection-oriented), ví dụ TCP
* phi kết nối (connectionless), ví dụ UDP

Tầng giao vận có thể được xem như một cơ chế vận chuyển thông thường, nghĩa là trách nhiệm của một phương tiện vận tải là đảm bảo rằng hàng hóa/hành khách của nó đến đích an toàn và đầy đủ.Tầng giao vận cung cấp dịch vụ kết nối các ứng dụng với nhau thông qua việc sử dụng các cổng TCP và UDP. Do IP chỉ cung cấp dịch vụ phát chuyển nỗ lực tối đa (best effort delivery), tầng giao vận là tầng đầu tiên giải quyết vấn đề độ tin cậy.

***Tầng mạng(Network):*** IP (IPv4, IPv6) ARP (Address Resolution Protocol| - tạm dịch là Giao thức phân giải địa chỉ) và RARP (Reverse Address Resolution Protocol - tạm dịch là Giao thức tìm địa chỉ ngược lại) hoạt động ở bên dưới IP nhưng ở trên tầng liên kết (link layer), vậy có thể nói là nó nằm ở khoảng trung gian giữa hai tầng.

Tầng mạng giải quyết các vấn đề dẫn các gói tin qua một mạng đơn. Một số ví dụ về các giao thức như vậy là X.25, và giao thức Host/IMP của mạng ARPANET.

Trong bộ giao thức liên mạng, giao thức IP thực hiện nhiệm vụ cơ bản dẫn đường dữ liệu từ nguồn tới đích. IP có thể chuyển dữ liệu theo yêu cầu của nhiều giao thức tầng trên khác nhau; mỗi giao thức trong đó được định danh bởi một số hiệu giao thức duy nhất: giao thức ICMP (Internet Control Message Protocol) là giao thức 1 và giao thức IGMP (Internet Group Management Protocol) là giao thức 2.

Một số giao thức truyền bởi IP, chẳng hạn ICMP (dùng để gửi thông tin chẩn đoán về truyền dữ liệu bằng IP) và IGMP (dùng để quản lý dữ liệu đa truyền (multicast)), được đặt lên trên IP nhưng thực hiện các chức năng của tầng liên mạng, điều này minh họa một sự bất tương thích giữa liên mạng và chồng TCP/IP và mô hình OSI. Tất cả các giao thức định tuyến, chẳng hạn giao thức BGP (Border Gateway Protocol), giao thức OSPF, và giao thức RIP (Routing information protocol|), đều thực sự là một phần của tầng mạng, mặc dù chúng có thể có vẻ thuộc về phần trên của chồng giao thức.

***Tầng liên kết(Data link):*** Ethernet, Wi-Fi, Token ring, PPP, SLIP, FDDI, ATM, Frame Relay, SMDS, … Những tầng gần trên nóc gần với người sử dụng hơn, còn những tầng gần đáy gần với thiết bị truyền thông dữ liệu. Mỗi tầng có một giao thức để phục vụ tầng trên nó, và một giao thức để sử dụng dịch vụ của tầng dưới nó (ngoại trừ giao thức của tầng đỉnh và tầng đáy).

Các giao thức thuộc tầng liên kết được sử dụng để chuyển các gói tin từ tầng mạng tới các máy chủ (host) khác nhau. Các quá trình truyền các gói tin trên một liên kết cho trước và nhận các gói tin từ một liên kết cho trước có thể được điều khiển cả trong phần mềm điều vận thiết bị (device driver) dành cho card mạng, cũng như trong firmware hay các chipset chuyên dụng. Những thứ đó sẽ thực hiện các chức năng liên kết dữ liệu chẳng hạn như bổ sung một tín đầu (packet header) để chuẩn bị cho việc truyền gói tin đó, rồi thực sự truyền frame dữ liệu qua một môi trường vật lý [11].

Tầng liên kết còn có thể là tầng nơi các gói tin được chặn (intercepted) để gửi qua một mạng riêng ảo (virtual private network). Khi xong việc, dữ liệu tầng liên kết được coi là dữ liệu của ứng dụng và tiếp tục đi xuống theo chồng giao thức TCP/IP để được thực sự truyền đi. Tại đầu nhận, dữ liệu đi lên theo chồng TCP/IP hai lần (một lần cho mạng riêng ảo và lần thứ hai cho việc định tuyến).

Tầng liên kết còn có thể được xem là bao gồm cả tầng vật lý - tầng là kết hợp của các thành phần mạng vật lý thực sự (hub, các bộ lặp (repeater), cáp mạng, cáp quang, cáp đồng trục (coaxial cable), card mạng, cạc HBA (Host Bus Adapter) và các thiết bị nối mạng có liên quan: RJ-45, BNC, etc), và các đặc tả mức thấp về các tín hiệu (mức hiệu điện thế, tần số, v.v..).

## Kĩ thuật lập trình mạng

Kĩ thuật lập trình mạng là quá trình phát triển các ứng dụng mạng bằng cách sử dụng các công nghệ và giao thức mạng. Nó bao gồm việc tạo ra các ứng dụng mạng có tính năng như truyền tải dữ liệu, định tuyến, xử lý tín hiệu và xử lý lỗi.

Kĩ thuật lập trình mạng bao gồm các công nghệ như TCP/IP, HTTP, FTP, DNS và các ngôn ngữ lập trình như C, C++, Java, Python và Ruby. Kỹ năng lập trình mạng cần thiết để phát triển các ứng dụng mạng phức tạp như hệ thống mạng, ứng dụng web, ứng dụng di động, game đa người chơi trực tuyến, và nhiều ứng dụng khác.

Các kỹ năng cần thiết để trở thành một lập trình viên mạng bao gồm hiểu biết về cấu trúc mạng, các giao thức mạng, quản lý tài nguyên mạng và bảo mật mạng. Ngoài ra, lập trình viên mạng cũng cần có khả năng làm việc với các thư viện và các công cụ phát triển phần mềm để tạo ra các ứng dụng mạng chất lượng cao.

Bài báo cáo này hướng đến kĩ thuật lập trình mạng ngôn ngữ chính là C++, trên Qt Creator. Qt Creator là một môi trường phát triển tích hợp (IDE) được thiết kế để phát triển ứng dụng sử dụng Qt Framework. Nó được phát triển bởi Qt Company và cung cấp cho các lập trình viên một công cụ để tạo và triển khai các ứng dụng đa nền tảng.

Qt Creator cung cấp cho người dùng các tính năng như trình soạn thảo mã nguồn, kiểm tra cú pháp, dự đoán từ khóa, xây dựng, debug và mô phỏng ứng dụng. Nó cũng có khả năng hiển thị giao diện người dùng, cho phép người dùng tạo và sửa đổi các đối tượng UI như cửa sổ, nút, thanh trượt, biểu đồ, v.v.

Ngoài ra, Qt Creator còn hỗ trợ các tính năng như auto-complete, refactoring, kiểm tra lỗi trên nhiều nền tảng và tích hợp các công cụ quản lý phiên bản như Git và Subversion. Qt Creator được cung cấp miễn phí và được phát triển trên nhiều nền tảng, bao gồm Windows, macOS và Linux.

Để giao tiếp mạng trên Qt với ngôn ngữ C++ ta cần đáp ứng 3 điều kiện:

* Biết địa chỉ IP của máy tính đối phương.
* Có 1 cổng mạng (port) mở và chưa được sử dụng.
* Sử dụng cùng 1 giao thức truyền dữ liệu.

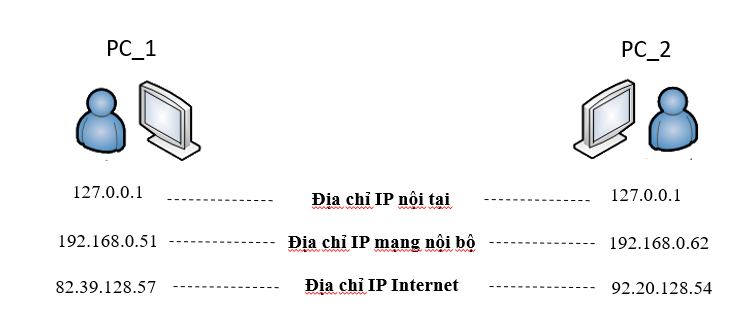
Địa chỉ IP, định danh của máy tính trong mạng là nơi giúp các máy tính nhận ra nhau trong cả mạng lưới.

Về cơ bản, chúng ta có thể xác định được là mỗi máy tính có trung bình 3 địa chỉ IP:

**Địa chỉ IP nội tại:** còn được gọi là địa chỉ localhost hay loopback. Địa chỉ này được sử dụng bởi bản thân mỗi chiếc máy tính để chỉ chính bản thân mình. Nó không quá hữu dụng do không có dính dáng gì đến mạng lưới cả. Thế nhưng lại khá có ích khi chúng ta cần dùng để chạy thử chương trình. Ví dụ 127.0.0.1.

**Địa chỉ IP mạng nội bộ:** Nếu có đồng thời nhiều máy tính cùng kết nối vào mạng ở nhà riêng thì nhờ địa chỉ IP này, chúng có thể liên lạc với nhau mà không cần thông qua mạng Internet. Ví dụ 192.168.0.51.

**Địa chỉ IP Internet:** Đây là địa chỉ dùng để liên lạc với tất cả các máy tính khác trên mạng Internet khắp thế giới. Ví dụ 82.39.128.57.



Hình 5. Hình minh hoạ giao tiếp qua các địa chỉ IP

Địa chỉ IP của máy tính gồm:

**IP nội tại:** mặc định là 127.0.0.1 hoặc cụm từ « localhost »

**IP mạng nội bộ:** tùy thuộc vào hệ điều hành mà chúng ta sử dụng :

* Trong Windows , hãy mở cửa sổ dòng lệnh (nơi chúng ta vẫn hay dùng để gõ các dòng lệnh Qt) và gõ câu lệnh sau: **ipconfig**
* Trong Linux hoặc Mac OS, tương tự nhưng với câu lệnh hơi khác: **ifconfig**

**IP trên Internet :** tra cứu trên các trang web cho phép cung cấp thông tin về IP.

Sau khi đã xác định được địa chỉ IP của người đối thoại, chúng ta có thể gần như liên lạc với người đó rồi. Chỉ còn thiếu thông tin về cổng liên hệ với máy tính đối phương là được. Vậy nên thuật ngữ chúng ta sẽ tiếp xúc tiếp theo là cổng(port) [12].

Khi máy tính kết nối vào mạng, ở mỗi thời điểm đều nhận đồng thời vô số tin nhắn khác nhau. Hoàn toàn bình thường khi chúng ta vừa có thể lướt web vừa đồng thời có thể tra cứu thư điện tử và để quản những thao tác đó người ta tạo ra khái niệm cổng.

1 cổng là 1 số nguyên nằm giữa 0 và 65535. Dưới đây là 1 số các cổng nổi tiếng :

* 21: sử dụng bởi các chương trình FTP (File Transfer Protocol) chuyên gửi nhận tệp.
* 80: cổng xử lý các yêu của trang web, hay được dùng bởi Firefox hay BanDoX.
* 110: dùng để nhận thư.

Phần lớn các cổng có số nhỏ hơn 1024 đều đã được đặt sẵn trước bởi máy tính. Thế nên chúng ta cần viết chương trình ưu tiên sử dụng các cổng từ 1024 đến 65535. Để liên lạc với máy tính khác.

Để tránh việc bất cứ chương trình nào trong mạng cũng có thể liên lạc và truy cập đến các máy tính mà không cần sự cho phép, tưởng lửa(Firewall) được ra đời. Nhiệm vụ của tường lửa là chặn hầu hết tất các các cổng và chỉ mở 1 số cổng được đánh giá là an toàn. Vậy nên trước khi lập trình thì chúng ta cần xem xét kỹ tùy chỉnh của tường lửa có thể đã được thiết lập trên máy của chúng ta từ trước vì nó có thể chặn cả các kết nối đến máy của chúng ta.

Sau khi có địa chỉ IP và các cổng. Địa chỉ IP thì cần tìm ra còn cổng sử thì cần chọn từ các cổng trống có sẵn. chúng ta đã có thể tạo ra 1 kết nối từ xa tới 1 chiếc máy tính khác trên mạng. để thực hiên điều đó ta cần sử dụng cùng 1 giao thức. giao thức là 1 tập hợp các quy tắc cho phép 2 máy tính trao đổi với nhau. 2 máy tính bắt buộc phải sử dụng cùng 1 giao thức thì mới có thể trao đổi được dữ liệu.

Có tồn tại hàng trăm giao thức liên lạc khác nhau. Chúng có thể rất phức tạp nhưng cũng có khi rất đơn giản, tùy theo cấp độ trao đổi « cao » hay « thấp ». Chúng ta có thể chia chúng thành 2 loại lớn :

* **Giao thức cấp độ cao:** ví dụ như giao thức FTP sử dụng cổng 21 để gửi và nhận tệp tin là 1 hệ thống trao đổi dữ liệu ở cấp độ cao. Cách thức hoạt động của nó đã được miêu tả rất rõ ràng. Vậy nên giao thức này khá dễ sử dụng nhưng trái lại, chúng ta khó có thể thêm vào các chỉnh sửa mới.
* **Giao thức cấp độ thấp:** ví dụ như giao thức TCP. Nó được sử dụng bởi các chương trình mà không giao thức cấp độ cao nào thích hợp với chúng. Khi sử dụng giao thức này, chúng ta có thể sẽ phải thao tác với từng đơn vị dữ liệu nhỏ nhất. Vậy nên đương nhiên là nó khó sử dụng hơn FTP nhưng chúng ta có thể làm tất cả những gì chúng ta muốn với nó.

Dữ liệu trong mạng thường không thể truyền tải 1 lúc quá nhiều mà bắt buộc phải chia ra thành những thành phần nhỏ hơn mà chúng ta gọi là gói. Mỗi gói lại có thể được chia thành các gói nhỏ hơn.

Các gói nhỏ có thể được gửi đi theo các cách khác nhau tùy theo giao thức cấp thấp mà chúng ta sử dụng:

* **Giao thức TCP:** giao thức thuộc loại cổ điển. Nó cần phải thiết lập 1 đường truyền giữa 2 máy trước khi dữ liệu được chuyển đi. Hệ thống quản lý gói tin được thiết lập để phòng trường hợp 1 trong số các gói tin bị thất lạc và hệ thống sẽ gửi lại 1 gói khác thay thế. Vậy nên, với TCP chúng ta có thể chắc chắn là mọi gói tin đều được gửi tới nơi và theo đúng thứ tự. Trái lại, hệ quả là của việc này là khiến cho dữ liệu được truyền đi chậm hơn so với sử dụng UDP.
* **Giao thức UDP:** giao thức này không cần thiết lập trước đường truyền và nó truyền dữ liệu đi nhanh hơn. Trái lại, do các gói tin không được quản lý, việc các gói tin bị thất lạc có thể xảy ra và chúng ta không thể xác định được. 1 khả năng xấu khác là các gói tin có thể đều đến nơi nhưng theo thứ tự lộn xộn.

Đối với lập trình mạng trong Qt các kĩ thuật cơ bản cần có gồm:

**Xử lý kết nối mạng:** Sử dụng các lớp QTcpSocket hoặc QUdpSocket để tạo kết nối mạng và truyền dữ liệu giữa các ứng dụng. Lập trình viên có thể sử dụng các phương thức để thiết lập kết nối, gửi và nhận dữ liệu qua mạng.

**Xử lý các giao thức mạng:** Sử dụng các lớp QTcpServer để tạo server và lắng nghe các kết nối đến từ các client. Lập trình viên có thể sử dụng các phương thức để đọc và ghi dữ liệu vào các socket để giao tiếp với client. Các giao thức mạng phổ biến như HTTP, FTP, Telnet và SSH cũng có thể được xử lý bằng cách sử dụng các thư viện hoặc các lớp được cung cấp sẵn trong Qt.

**Xử lý đa luồng:** Sử dụng các lớp QThread hoặc QThreadPool để xử lý đa luồng trong ứng dụng mạng. Lập trình viên có thể sử dụng các phương thức để tạo và quản lý các thread trong ứng dụng để xử lý các tác vụ đồng thời.

**Xử lý giao diện người dùng:** Sử dụng các lớp QWidget hoặc QML để tạo giao diện người dùng cho ứng dụng mạng. Lập trình viên có thể sử dụng các phương thức để tạo và quản lý các đối tượng UI như cửa sổ, nút, thanh trượt, biểu đồ, v.v. và kết nối chúng với các tác vụ mạng.

**Xử lý lỗi mạng:** Trong quá trình phát triển ứng dụng mạng trên Qt, lập trình viên cần phải xử lý các lỗi mạng có thể xảy ra, bao gồm mất kết nối, timeout, và các lỗi mạng khác. Qt cung cấp các cơ chế để xử lý các lỗi này, như các sự kiện kết nối, sự kiện đọc/ghi, và các phương thức để xử lý lỗi.

**Sử dụng các công cụ hỗ trợ:** Qt Creator là một công cụ phát triển tích hợp cho Qt, cung cấp một môi trường phát triển tích hợp (IDE) để phát triển ứng dụng mạng trên Qt. Nó cung cấp các tính năng hỗ trợ như kiểm tra cú pháp, gỡ rối, xây dựng và triển khai ứng dụng. Ngoài ra, Qt cũng cung cấp các công cụ hỗ trợ khác như Qt Linguist để quản lý đa ngôn ngữ trong ứng dụng, Qt Test để kiểm tra và xác minh tính chính xác của ứng dụng, và Qt Installer Framework để tạo trình cài đặt cho ứng dụng.

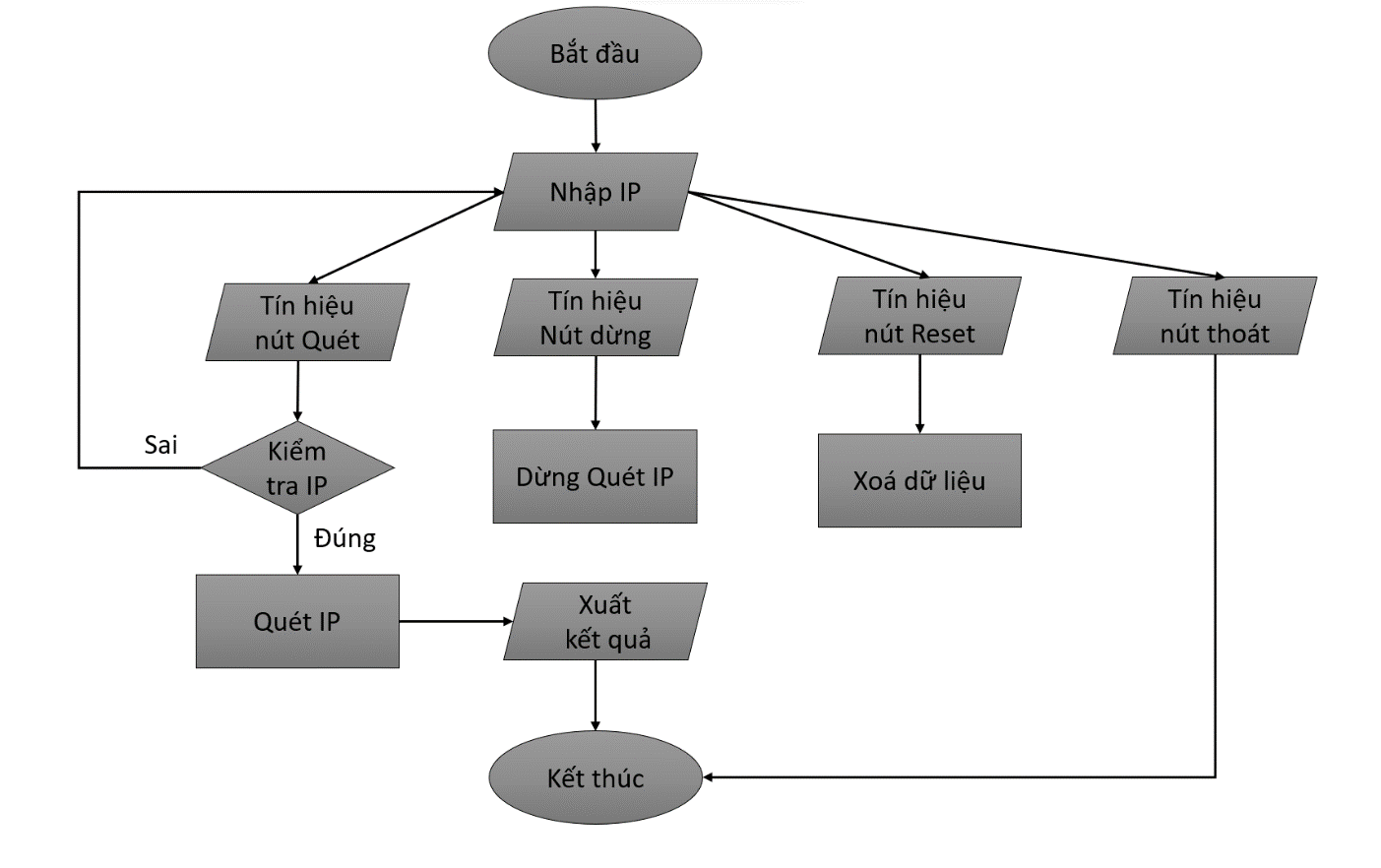
# CHƯƠNG 3.

# ĐEMO VÀ ĐÁNH GIÁ NGUY CƠ BẢO MẬT

## Thực hiện xây dựng chương trình

Thực hiện xây dựng chương trình trên Qt Creator:

Xây dựng cấu trúc trúc thuật toán như sau:



Hình 6. Sơ đồ cấu trúc thuật toán chương trình

Chương trình bao gồm:

**Nhập IP:** nơi nhập địa chỉ IP mà chúng ta cần quét cổng mạng.

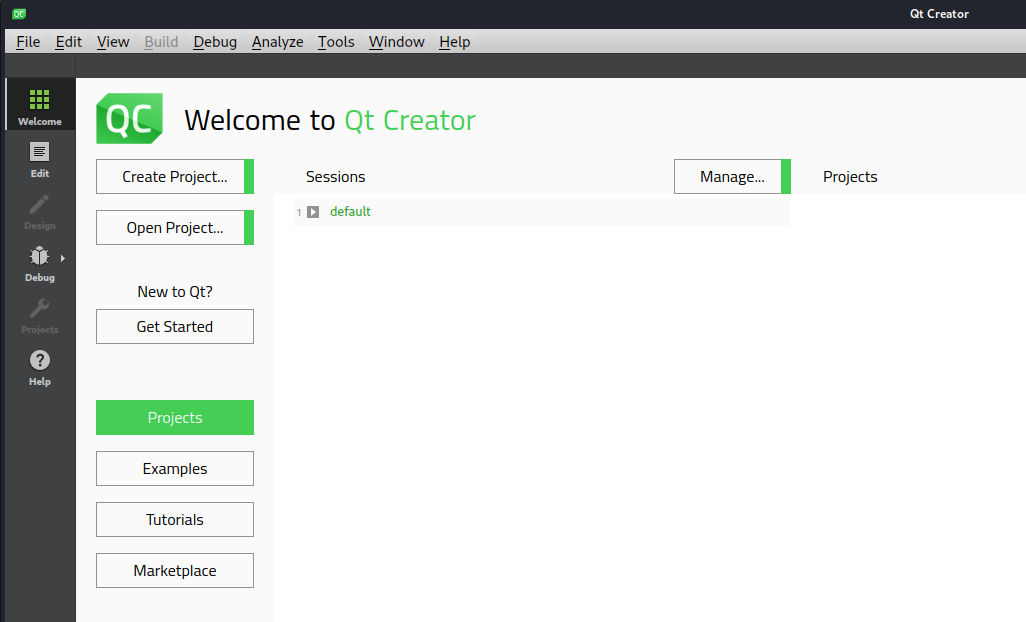
Có 4 tín hiệu chính gồm 4 nút:

* **Nút Quét:** Thực hiện quét cổng trên IP hiện tại mà ta nhập vào.
* **Nút Dừng:** Dừng quá trình quét cổng hiện tại đang quét.
* **Nút Reset:** Để xoá dữ liệu IP hiện tại và kết quả hiện tại.
* **Nút Thoát:** Thoát chương trình.

**Hàm kiểm tra IP:** Kiểm tra IP hiện tại có hợp lệ, trong chương trình chúng ta sẽ kiểm tra IP hợp lệ là Ipv4.

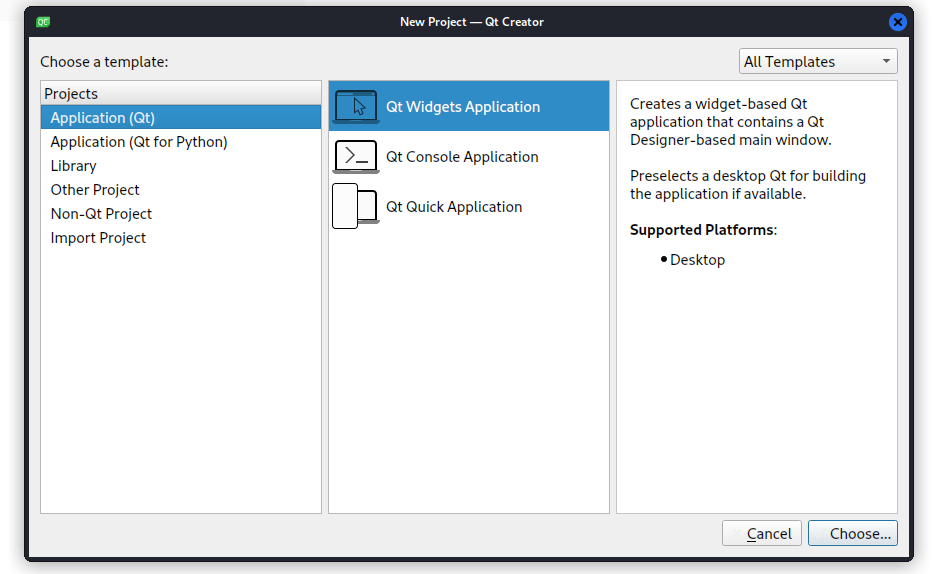
**Xuất kết quả:** hiện thị kết quả thu được sau quá trình quét cổng mạng hoàn tất và xuất ra file.

### Tạo project trên Qt Creator



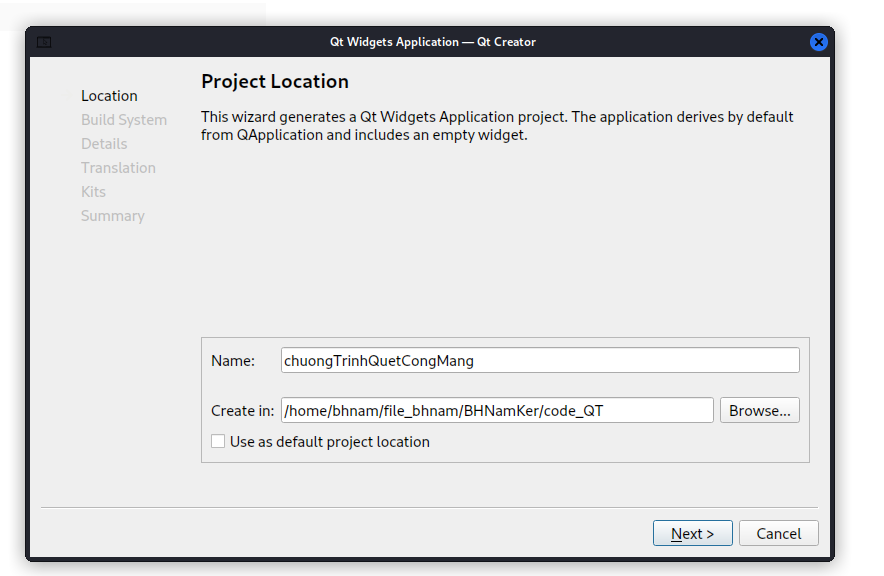
Hình 7. Giao diện Qt Creator

Trên hệ điều hành Linux tài đặt ứng dụng Qt Creator sau khi cài đặt chúng ta tiến hành tạo project.



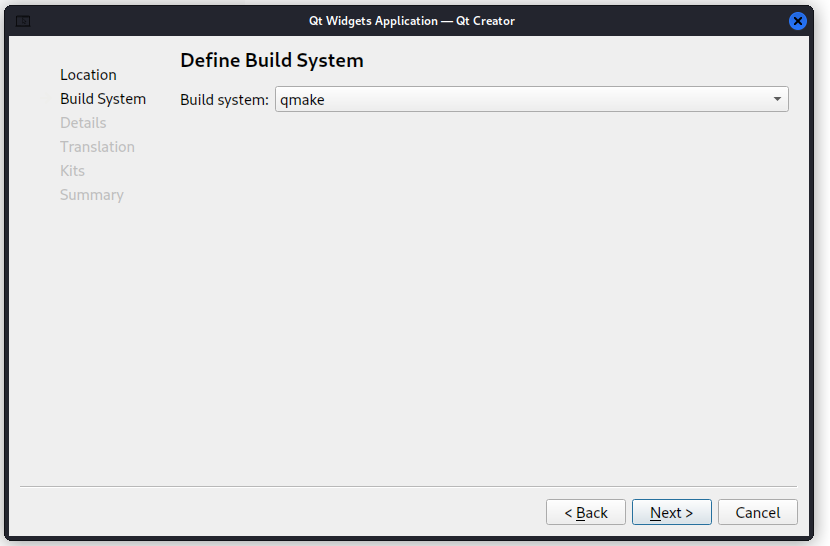
Hình 8. Tạo Qt Widget Application

Tạo Project Application chọn vào Qt Widgets Application đển tạo ra các Widgets giao diện chương trình là các Widgets.



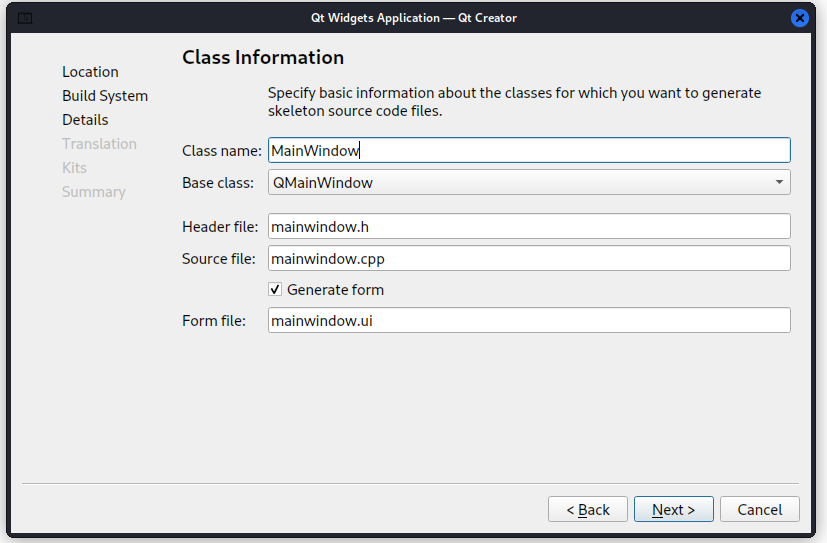
Hình 9. Tạo Project Location

Trong phần Project Location, chúng ta thực hiện đặt tên cho project là chươngTrinhQuetCongMang, và đường dẫn tạo project tuỳ thuộc vào máy người dùng và chúng ta có thể click vào bên dưới để sử dụng đường dẫn này dùng làm đường dẫn mặc định cho các project sau khi tạo.



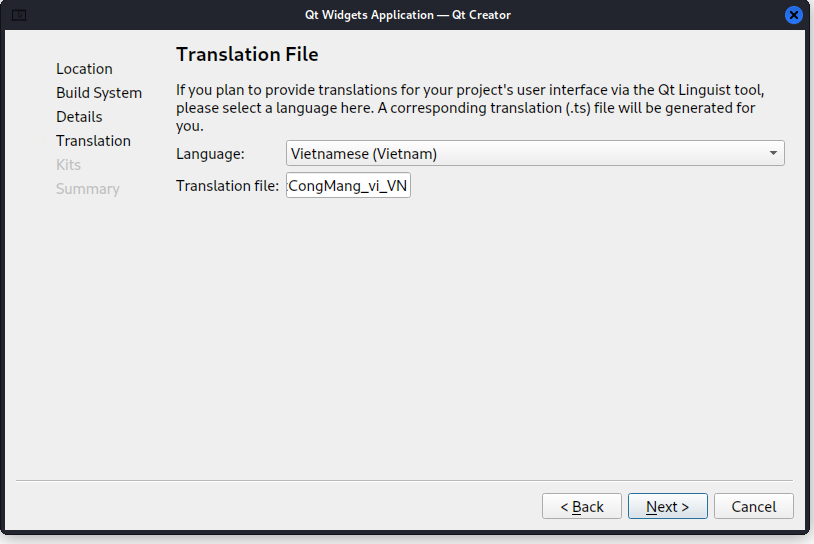
Hình 10. Tạo Define Build System

Chọn build system là qmake.



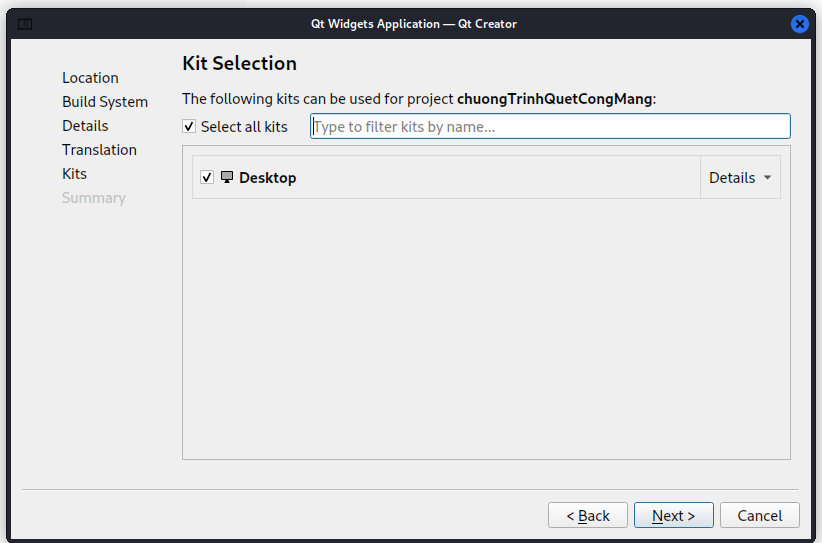
Hình 11. Tạo Class information

Trên Qt Creator tạo mặc định cho chúng ta các class trong phần Class Information, chúng ta chọn next.



Hình 12. Tạo Translation File

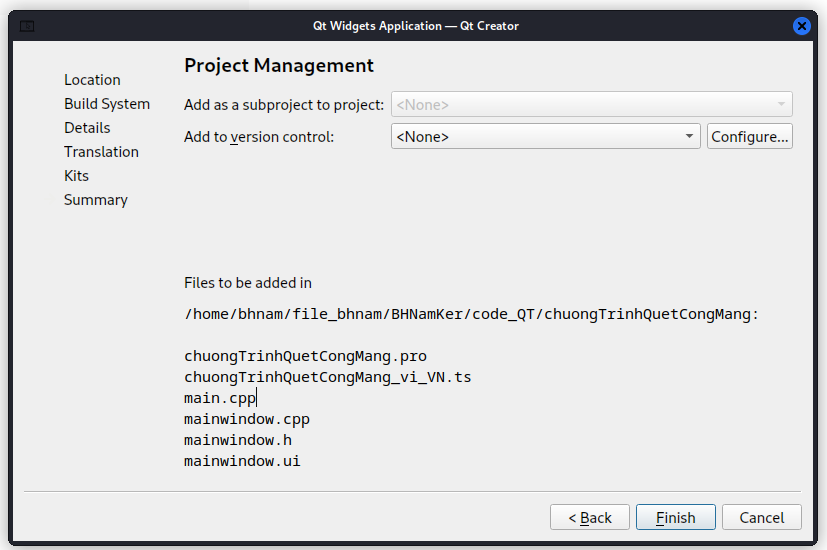
Trong phần Translation File chúng ta chọn ngôn ngữ là Việt Nam rồi ấn next.



Hình 13. Tạo kit selection

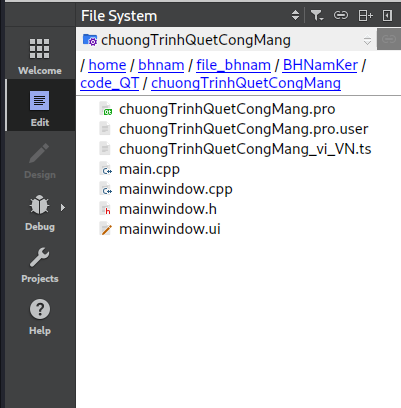
Trong Kit selection ta tích và Desktop và chọn next.

Cuối cùng là chúng ta được các thông tin sau:



Hình 14. Kết quả Project

Chọn Finish để hoàn thành tạo project, Ta được cấu trúc như sau:



Hình 15. File system của chương trình

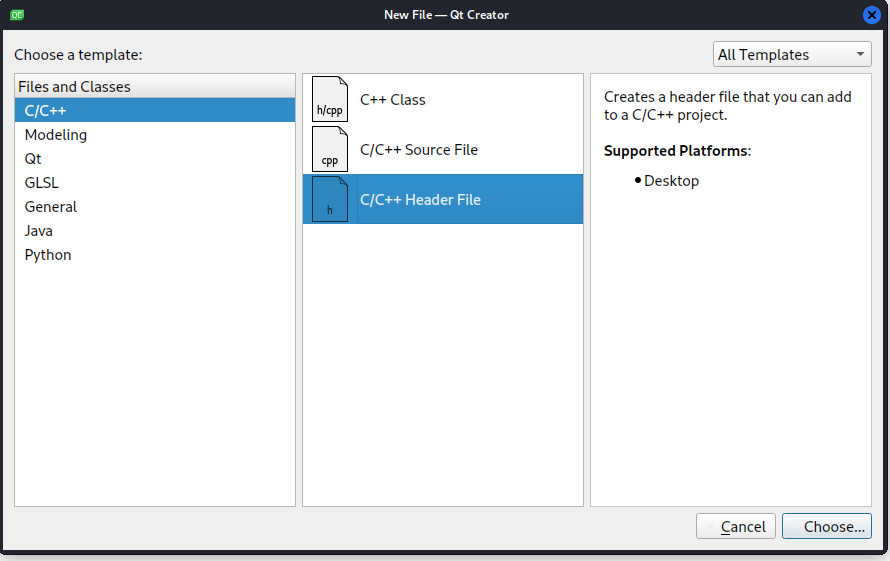
Để tạo chương trình quét cổng chúng ta quan tâm chính đến những file sau:

File main.cpp, file chuongTrinhQuetCongMang.pro và hai file .h và .cpp để tạo giao diện và các hàm xử lý trên chương trình.

### Tạo giao diện chương trình Quét cổng

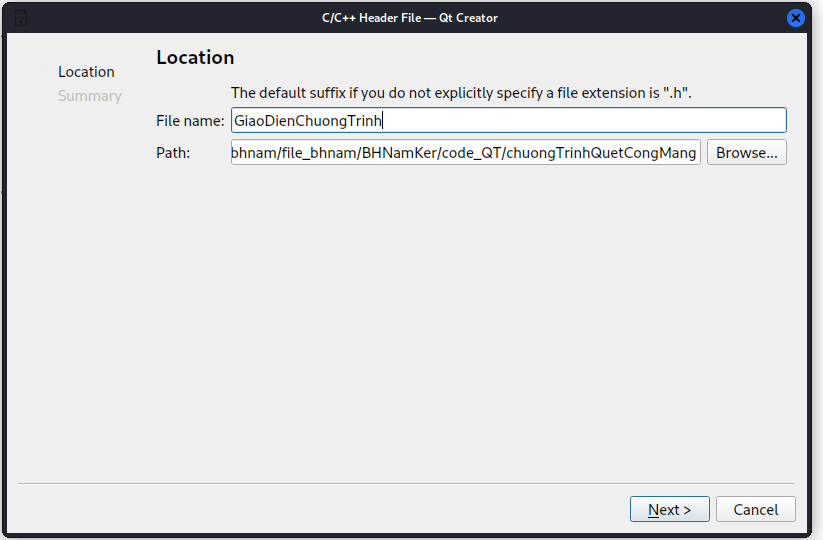
Như đã nhắc đến ở phần trước điều chúng ta quan tâm đến chính là file.pro, file.cpp và file.h.

Tạo sẽ tạo file.h có tên là GiaoDienCuongTrinh.h, trong phần project ta nhấn chột phải tìm đến phần new file và chọn C/C++ Header File.

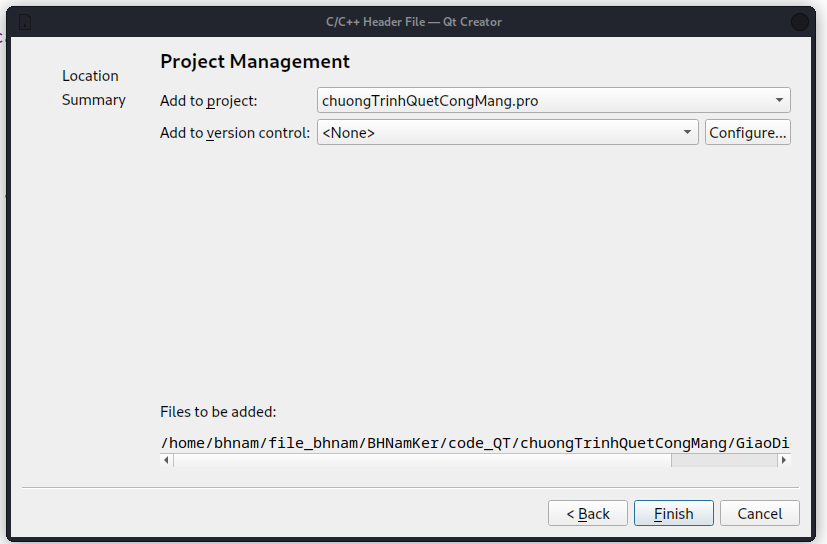


Hình 16. Tạo file GiaoDienChuongTrinh.h

Path dẫn ở đây ta để mặc định dẫn đến project.

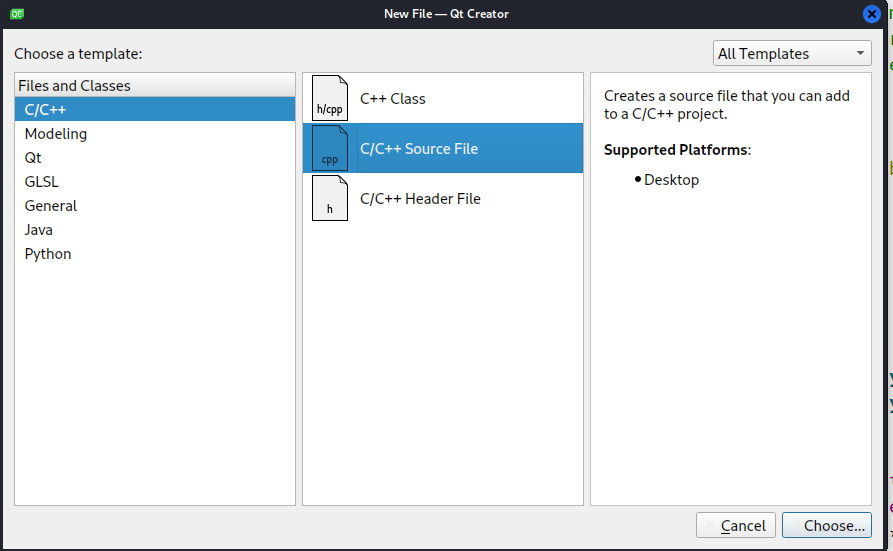


Hình 17. Location file GiaoDienChuongTrinh.h

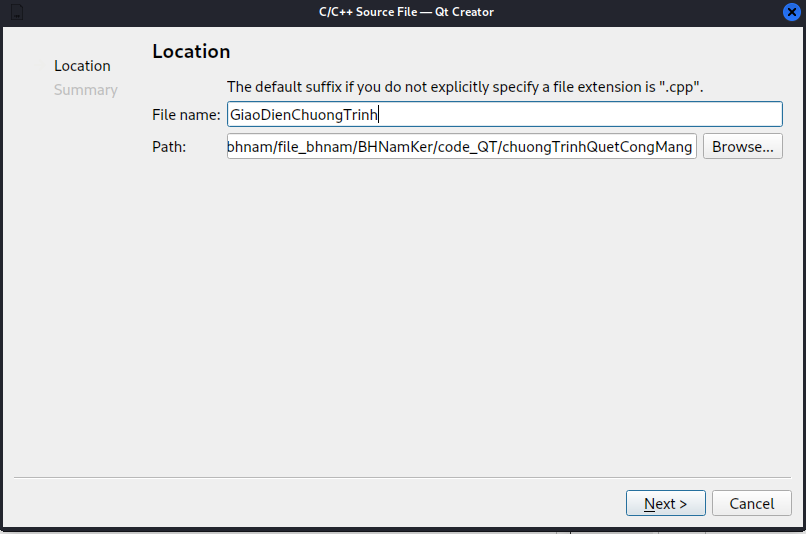


Hình 18. qmake file GiaoDienChuongTrinh.h vào chương trình

Chọn Finish để hoàn thành tạo file. Sau đó ta tiếp tục tạo file.cpp đặt tên là GiaoDienChuongTrinh.cpp.

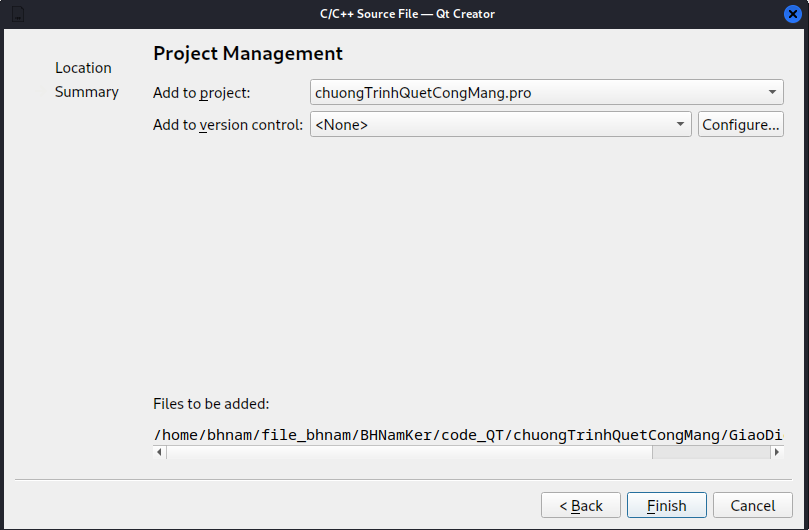


Hình 19. Tạo file GiaoDienChuongTrinh.cpp



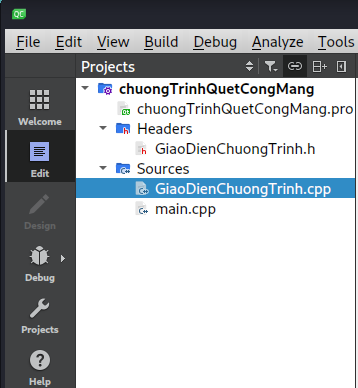
Hình 20. Tạo location file GiaoDienChuongTrinh.cpp

Chọn next đến phần Project Management và chọn Finish để hoàn thành tạo file.cpp.



Hình 21. Qmake file GiaoDienChuongTrinh.cpp vào chương trình

Cấu trúc chương trình của chúng ta gồm như sau:

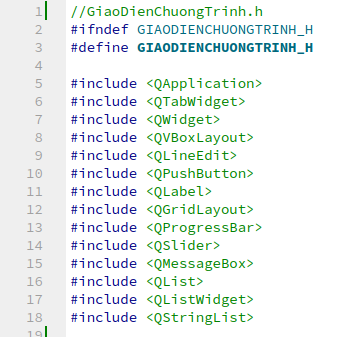


Hình 22. Project chuongTrinhQuetCongMang

### Tạo các hàm xử lý giao diện

Trong phần tạo các hàm xử lý dữ liệu, chọn vào file GiaoDienChuongTrinh.h

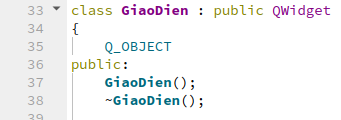
Thêm các thư viện cần thiết để chúng ta tạo ra một giao diện người dùng:



Hình 23. Các thư viện cần thiết tạo giao diện

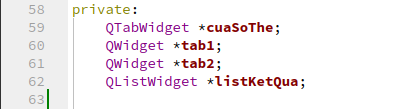
Trong Qt Creator hỗ trợ rất nhiều thư viện đã cấu hình cài đặt sẵn nên chúng ta chỉ cần kế thừa và sử dụng.

Tiếp theo ta sẽ tạo một class có tên là GiaoDien được kết thừa từ lớp Widget.



Hình 24. Tạo class GiaoDien

Gồm có phương thức khởi tạo là GiaoDien(); và huỷ là ~GiaoDien(); trong phần public.



Hình 25. Private class GiaoDien

Trong private chúng ta tạo ra các đối tượng con trỏ để xây dựng giao diện:

Con trỏ \*cuaSoThe là một TabWidget cho phép chúng ta tạo ta các tab giao diện trên chương trình. Con trỏ \*tab1 và \*tab2 là 2 đối tượng Widget sẽ được chúng ta tạo ra để set vào TabWidget. Và con trỏ đối tượng \*listKetQua giúp hiển thị kết quả khi thực hiện quét cổng mạng.

Chúng ta sẽ chuyển sang phần GiaoDienChuongTrinh.cpp để cấu thành các đối tượng trên.



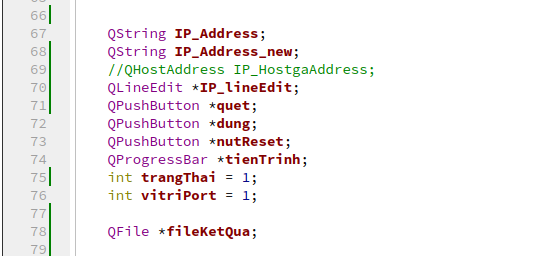
Hình 26. Khởi tạo class GiaoDien

Chúng ta sẽ thêm thư viện GiaoDienChuongTrinh.h qua dòng lệnh số 2.

Khởi tạo giao diện gồm cuaSoThe là TabWidget, đối tượng tab1 và tab2 là các Widget. Sau đó chúng ta sẽ thêm 2 đối tượng vào TabWidget và có tiêu đề tab1 là “Thong Tin”, tab2 là “Ket Qua”. Theo đó là gọi hàm create\_tab1(); và create\_tab2(); để tạo cấu tạo 2 tab. Hàm create\_fileKetQua(); sẽ được phân tích ở phần sau. Trong 3 dòng cuối ở hàm này là tạo một đối tượng con trỏ \*vbox thuộc VboxLayout và thêm đối tượng cuaSoThe mà chúng ta đã tạo vào trong. Cuối cùng là setLayout(vbox).

Qua lại với phần private trong file GiaoDienChuongTrinh.h

Chúng ta khởi tạo thêm các đối tượng sau:



Hình 27. Thêm các đối tượng và biến cho class GiaoDien

Đối tượng IP\_Address và IP\_Address\_new hai đối tượng này sẽ nhận các dữ liệu là string từ người dùng đăng nhập vào từ đối tượng con trỏ \*IP\_lineEdit.

Chúng ta sẽ tạo ra 3 đối tượng nút bấm trong chương trình đó là “quet”, “dung”, “nutReset”, và một con trỏ \*tienTrinh, thanh tiến trình thực hiện.

Biến trạng thái, vitriPort và con trỏ \*fileKetQua sẽ được phân tích đề cập khi xây dựng chương trình.

Sau đây là phần đầu của quá trình khởi tạo tab1.



Hình 28. Phần đầu của khởi tạo tab1

Chúng ta tạo ra đối tượng Label và Label2 để hiển thị “IP Address” và “Tien Trinh”

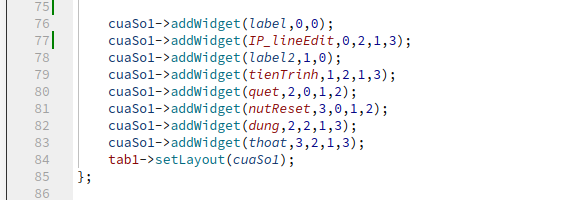
Thêm một nút bấm mới đó là thoat. Vì đối tượng này sẽ không tác động đến quá trình quét cổng của chúng ta nhiều nên sẽ không nằm trong phần private của class.

Tiếp theo là khởi tạo cho đối tượng nutReset, IP\_lineEdit, IP\_Address gán bằng kết quả của đối tượng IP\_lineEdit, đối tượng quet, dung là các nút bám PushButton. Đối tượng tienTrinh được set là 1 đến MAX.

MAX được chúng ta thêm vào file GiaoDienChuongTrinh.h là một hằng số.

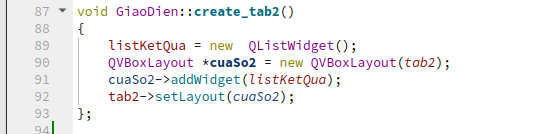
Hình 29. Biến hằng của chương trình

Vì tienTrinh sẽ hiện thì thông tin quét cổng cho chúng ta từng port 1 đến port tối đa là 65535. Giá trị ban đầu của tienTrinh là 1.



Hình 30. Thêm các đối tượng vào tab1

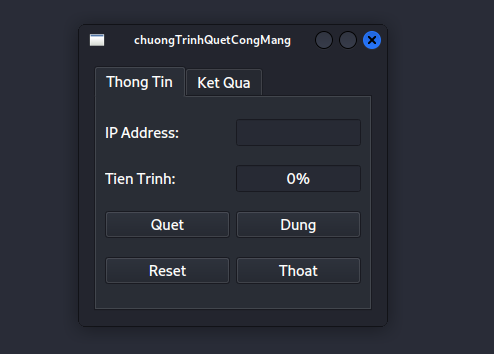
Cuối phần khởi tạo tab1 là quá trình thêm các đối tượng trên vào cuaSo1 với các vị trí thích hợp và setLayout cho tab1.



Hình 31. Khởi tạo tab2

Trong tab2 tạo đối tượng listKetQua có kiểu là ListWidget. cuaSo2 là VboxLayout sắp xếp đối tượng theo hàng và thêm đối tượng listKetQua và cuaSo2 và tab2 setLayout cuaSo2.

Sau khi hoàn thành các bước trên chúng ta chạy chương trình sẽ được kết quả:



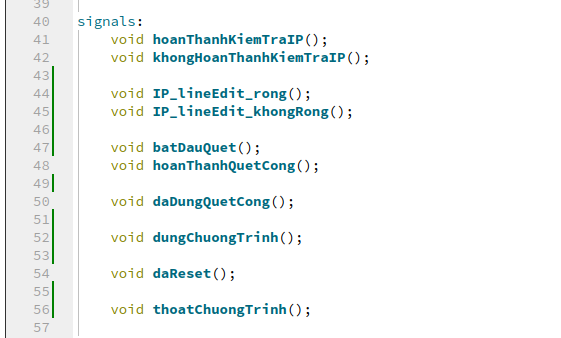
Hình 32. Kết quả giao diện

Với các bước thực hiện trên thì chúng ta đã tạo được một giao diện cho chương trình gồm nhập địa chỉ IP, thanh tiến trình, nút Quet cổng, nút Dung quá trình quét, nút Reset, và nut Thoat chương trình.

### Tạo các hàm xử lý tín hiệu

Để chương trình thực hiện nhập địa chỉ IP và nhấn các nút để quét, dừng, reset và thoát chương trình. Chúng ta tiến hành tạo các tín hiện xử lý cho chương trình trong file.h.

Chúng ta sẽ thêm các tín hiệu sau:



Hình 33. Các tín hiệu của chương trình

Tạo ra 2 tín hiệu **hoanThanhKiemTraIP**(), và **khongHoanThanhKiemTraIP**()

Hai tín hiệu này sẽ được dùng khi chúng ta thực hiện hàm kiểm tra IP nếu kiểm tra IP thành công thì chúng ta sẽ phát ra tín hiệu hoanThanhKiemTraIP() và kiểm tra IP không thành công sẽ phát ra tín hiệu khongHoanThanhKiemTraIP().

Hai tín hiệu tiếp theo là kiểm tra giá trị của đối tượng IP\_lineEdit trả về có hợp lệ hay không, đối tượng có giá trị rỗng và ngược lại.

Tín hiệu **batDauQuet**() sẽ được phát ra khi ta thực hiện hàm kiểm tra trạng thái.

Tín hiệu **hoanThanhQuetCong**() sẽ được phát ra khi chúng ta thực hiện hàm quét cổng hoàn tất. Tức là quét thành công địa chỉ IP hiện tại từ 1 đến MAX.

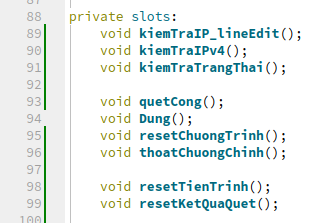
Tín hiệu **daDungQuetCong**() sẽ được phát khi chúng ta thực hiện dừng quét cổng

Tín hiệu **dungChuongTrinh**() sẽ được phát khi chúng ta thực hiện dung chương trình.

Tín hiệu **daReset**() sẽ được phát khi chúng ta thực hiện reset() chương trình.

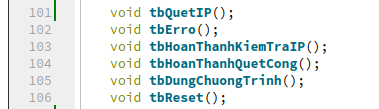
Tín hiệu **thoatChuongTrinh** sẽ phát ra khi chúng ta muốn thoát chương trình.

Chúng ta sẽ tạo các hàm slots sau:



Hình 34. Các SLOT của chương trình

Các hàm slot này sẽ được thực hiện khi được nhận các tín hiện.



Hình 35. Các SLOT thông báo

Các slot này là các hàm hiện thị thông báo khi được gọi tới.

Chúng ta sẽ tạo ra các sự kiện trong hàm khởi tạo tab1 sau đây:



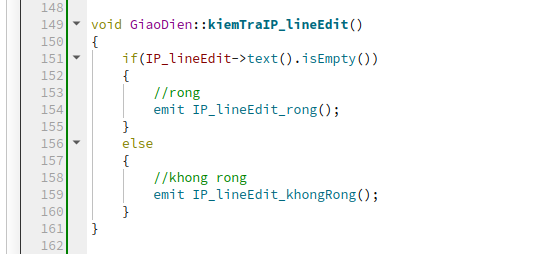
Hình 36. Các sự kiện tín hiệu

Chúng ta sẽ đi phân tích từng sự kiện trong các tín hiệu trên.

Tín hiệu đầu tiên nhận khi thực hiện nhấn nút quet:

Connect(quet,SIGNAL(clicked(bool)), this, SLOT(kiemTraIP\_lineEdit));

Khi chúng ta nhấn nút quét trên đối tượng quet qua phương thức click() sẽ là một tín hiệu, tại this ở đây là giao diện tab1, sau đó sẽ thực hiện hàm bên trong slot(). Gọi hàm kiemTraIP\_lineRdit();

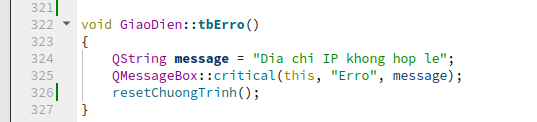


Hình 37. Hàm kiểm tra IP của lineEdit

Trong hàm này sẽ kiểm tra kết quả dữ liệu bên trong của đối tượng IP\_lineEdit kiểm tra xem có rỗng hay không nếu rỗng thì phát ta tín hiệu IP\_lineEdit\_rong() ngược lại phát ra tín hiệu IP\_lineEdit\_khongRong().

Sau khi thực hiện sau hàm kiemTraIP\_lineEdit. Chương trình sẽ nhận được 1 trong 2 tín hiệu hiệu của hàm. Nếu tín hiệu nhận được là rỗng chương trình sẽ thực hiện hàm tbErro() trong dòng lệnh số 55:

Connect(this, SIGNAL(IP\_lineEdit\_rong()), this, SLOT(tbErro()));



Hình 38. Hàm thông báo lỗi

Hàm tbErro() sẽ hiển thị lên màn hình một hộp thoại với tiêu đề “Erro” và nội dung là “Dia chi IP khong hop le” sao đó sẽ gọi hàm resetChuongTrinh() hàm này sẽ được chúng ta đề cập tới sau.

Nếu tín hiệu nhân vào là không rỗng chương trình sẽ thực hiện gọi hàm kiểm tra IPv4() thông qua dòng lệnh số 56:

Connect(this, SIGNAL(IP\_lineEdit\_khongRong()), this, SLOT(kiemTraIPv4()));



Hình 39. Hàm kiểm tra IPv4

Trong hàm này sẽ thực hiện gán giá trị hiện ta của đối tượng IP\_lineEdit cho IP\_Address\_new. Sau đó khởi tạo 1 đối tượng là octets sẽ chia string IP\_Address\_new thêm “.”. Nếu octets có size khác 4 là IP hiện tại đang không viết đúng định dạng của IPv4 và phát ra tín hiệu khongHoanThanhKiemTraIP();

Ngược lại, tiếp tục kiểm tra từng octets có giá trị từ 0 đến 255 không, nếu không sẽ phát ta tín hiệu khongHoanThanhKiemTraIP().

Phần này chúng ta đề cập đến biến trạng thái.

Chúng ta sẽ định nghĩa cho chương trình các trạng thái sau:

* 4 là trạng thái hoàn thành quét chương trình.
* 3 là trạng thái quét IP mới nhập vào.
* 2 là trạng thái đang quét IP thì có tín hiệu dừng.
* 1 là trạng thái bình thường để chạy quét chương trình. .
* 0 là trạng thái yêu cầu dừng quét.
* -1 là trạng thái yêu cầu reset chương trình.

Khi kiểm tra IP\_Address có rỗng không, nếu rỗng thì gán bằng IP\_Address\_new và biến trạng thái sẽ được gán bằng 1. Ngược lại kiểm tra IP\_Address có bằng IP\_Address\_new nếu bằng chúng ta sẽ kiểm tra trangThai bằng 4 không nếu bằng 4 thì chuyển trạng thái về 1, khác 4 chuyển trạng thái về 2. Cuối cùng, nếu IP\_Address không bằng IP\_Address\_new thì chuyển trạng thái về 3 gán IP\_Address bằng IP\_Address\_new.

Chương trình sau đó sẽ nhận được 2 tín hiệu:

* khongHoanThanhKiemTraIP(): sẽ gọi hàm tbErro().
* hoanThanhKiemTraIP(): sẽ gọi đến hàm kiemTraTrangThai().



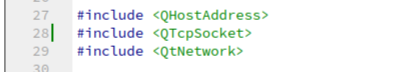
Hình 40. Hàm kiểm tra trạng thái chương trình

Hàm kiemTraTrangThai() sẽ thực hiện kiểm tra biến trangThai.

* **Trạng thái 1**: Thực hiện thông báo quét IP hiện tại, gọi hàm xoaDuLieuFile(), ghiIP() hàm này sẽ được đề cập sau. Đối tượng listKetQua sẽ được reset. Set lại vị trí port về 1. Phát ra tín hiệu quét.
* **Trạng thái 2**: Hiện hộp thoại hỏi người dùng có tiếp tục quét IP hiện tại. Nếu lựa chọn là Yes chúng ta chuyển trạng thái về 1 và phát tín hiệu quét. Lựa chọn là No, chúng ta sẽ cho quét IP lại từ đầu. Chuyển trạng thái về 1 và vị trí porot về 1, reset đối tượng listKetQua. Thông báo quét IP hiện tại gọi hàm xoaDuLieu() và ghiIP() sau đó phát tín hiệu quét.
* **Trạng thái 3**: Đang dừng quét thì nhập IP mới tương tự trạng thái 2 và phát tín hiệu quét.

Kết thúc hàm kiemTraTrangThai(), chương trình đa nhận được tín hiệu bắt đầu quét cổng.

Với hàm quét cổng ta sẽ thực hiện thêm các thư viện sau:

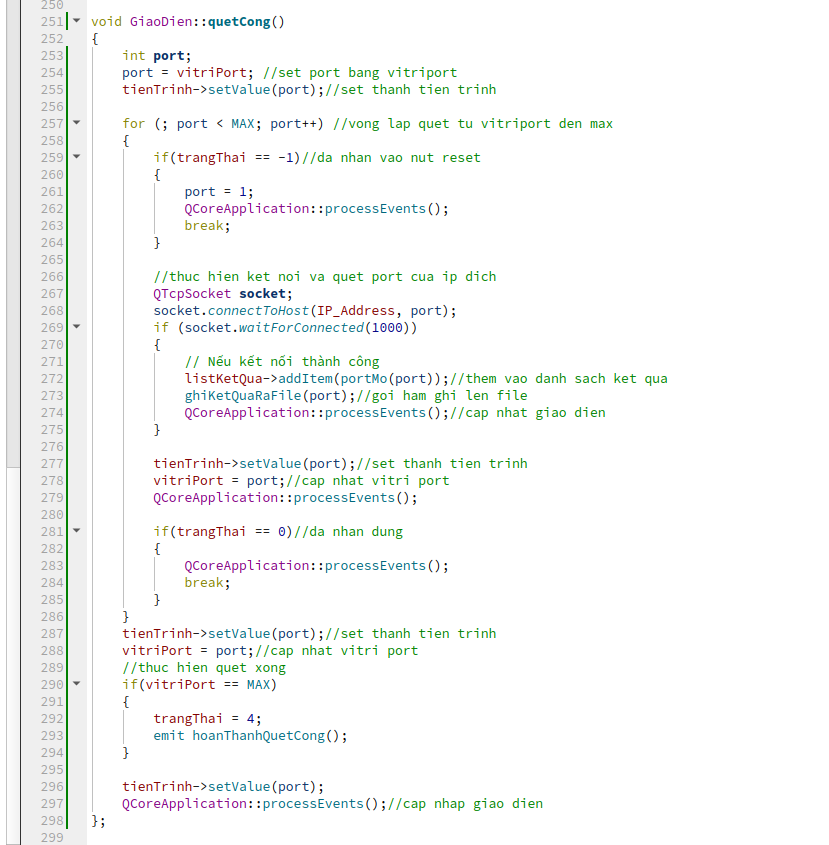


Hình 41. Thư viện cần cho hàm quét cổng

Thư viện QhostAddress: Lớp QHostAddress cung cấp các phương thức để biểu diễn và thao tác với các địa chỉ IP dưới dạng chuỗi ký tự hoặc giá trị số nguyên. Nó cũng cung cấp phương thức để kiểm tra tính hợp lệ của một địa chỉ IP và kiểm tra loại địa chỉ IP đó (IPv4 hay IPv6).

Thư viện QTcpSocket: để thiết lập kết nối giữa hai thiết bị thông qua mạng TCP/IP (Internet Protocol), cho phép truyền dữ liệu giữa các thiết bị thông qua mạng. Trong chương trình này chúng ta sẽ kết nối và đợi phải hồi khi kết nối được thiết lập thành công.

Thư viện QtNetwork: để truyền và nhận dữ liệu qua mạng, thực hiện các giao thức mạng (TCP, UDP, SSL/TLS, HTTP, FTP, ...).



Hình 42. Hàm quét cổng

Tạo biến port, gán giá trị bằng vị trí port, tienTrinh set giá trị bằng port.

Thực hiện vòng lặp quét cổng từ port đến MAX.

Trong vòng lặp chúng ta sẽ kiểm tra các trạng thái qua mỗi vòng lặp nếu trạng thái bằng -1 thì chúng ta cho port bằng 1.

Cập nhật giao diện qua lệnh QcoreApplication::processEvents().

Nếu trạng thái không thay đổi. Chúng ta tạo ra đối tượng socket và kết nối đến địa chỉ IP hiện tại với port hiện tại: socket.connectToHost(IP\_Address, port);

Kiểm tra trạng thái kết nối khi đợi sau 1s, qua lệnh socket.waitForConnected(1000)

Nếu trạng thái trả về True, kết nối thành công thực hiện ghi kết quả

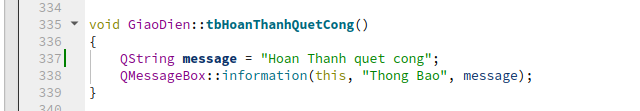
Sau đó thực hiện cập nhật giá trị cho thanh tiến trình cập nhật lại vị trí port.

Nếu trạng thái bằng 0, thoát khỏi vòng lặp.

Nếu thực hiện quét hết các cổng thì chuyển trang thái sau 4 và phát tín hiệu hoanThanhQuetCong(). Cập nhật lại thanh tiến trình và giao diện.

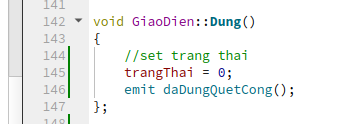
Chương trình nhận được tín hiệu hoàn thành quá trình quét cổng sau đó sẽ gọi đến hàm tbHoanThanhQuetCong() qua câu lệnh:

Connect(this,SIGNAL(hoanThanhQuetCong()),this,SLOT(tbHoanThanhQuetCon()));



Hình 43. Hàm thông báo quét cổng

Chương trình sẽ nhận được tín hiệu Dừng khi người người click vào nút Dung.



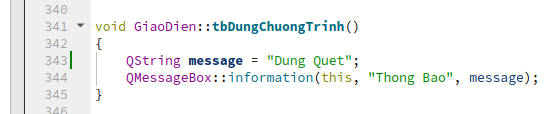
Hình 44. Hàm Dừng chương trình

Set trạng thái về 0 và phát tán hiệu dừng quét cổng.

Nhận được tín hiệu dừng quét cổng chương trình sẽ gọi hàm tbDungChuongTrinh().

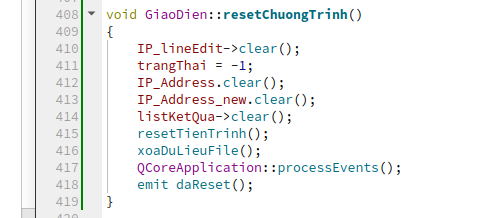
Qua lệnh:

Connect(this,SIGNAL(daDungQuetCong()),this,SLOT(tbDungChuongTrinh()));



Hình 45. Hàm thông báo Dừng chương trình

Nhận tín hiệu reset khi nhấn nút reset và gọi hàm resetChuongTrinh();

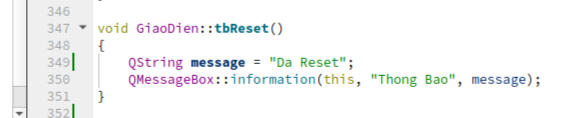


Hình 46. Hàm reset chương trình

Trong hàm này chúng ta xoá tất cả dữ liệu hiện sẵn trên chương trình.

Set trạng thái về -1, xoá IP\_Address, IP\_Address\_new, listKetQua, gọi hàm resetTienTrinh(), hàm xoaDuLieuFile(), cập nhật giao diện và phát tín hiệu daReset().

Nhận được tín hiệu đã reset chương trình thực hiện gọi hàm tbReset().



Hình 47. Hàm thông báo reset

Chương trình sẽ gọi hàm thoatChuongTrinh() khi người dùng click vào nút thoát. 

Hình 48. Hàm thoát chương trình

Trong hàm thoatChuongTrinh() chúng ta sẽ hỏi người dùng có muốn thoát không?. Nếu người dùng chọn Yes chúng ta sẽ chuyển trạng thái về 0 và gọi hàm qApp->exit để thoát chương trình, ngược lại sẽ phát tín hiệu dungChuongTrinh().

Chương trình nhận tín hiệu dungChuongTrinh() sẽ gọi hàm Dung() qua lệnh:

Connect(this, SIGNAL(dungChuongTrinh()),this,SLOT(Dung()));

### Tạo hàm xuất dữ liệu và kết quả

Chương trình sẽ tạo ra một thư mục và một file mỗi lần hoàn thành quét cổng.



Hình 49. Các đối tượng tạo thư mục, file và hàm xử lý

Chúng ta sẽ tạo ra các hàm thông báo: tbLoiTaoThuMuc(), tbLoiTaoFile().

Hàm xoaDuLieuFile() để xoá toàn bộ dữ liệu mà chúng ta sẽ ghi tra file.

Hàm kiemTraThucMucTonTai(Qstring tenThucMuc) tham số truyền vào là một string tên thư mục,

Một con trỏ đối tượng portMo(int port) truyền vào là một biến kiểu int. hàm này được sử dụng khi mỗi lần kết quả kết nối thành công.

Hàm ghiKetQuaRaFile(int port) là hàm để ghi kết quả ra file với tham số truyền vào là biến kiểu int.

Hàm ghiIP() là hàm sẽ ghi IP hiện tại lên file.

Quay trở lại hàm create\_fileKetQua():



Hình 50. Hàm khởi tạo file kết quả

Trong hàm này chúng ta tạo ra một string có tên là “ketQua” và kiểm tra tên này có tồn tại không. Nếu chưa tồn tại chúng ta tạo một thư mục có tên là “ketQua” và kiểm tra có tạo thành công không?. Nếu không thành công phát ra tín hiệu tbLoiTaoThuMuc().

Ngược lại con trỏ fileKetQua sẽ được tạo trong thư mục này có tên là “portOpen.txt”

Sau đó chúng ta sẽ mở file này nên qua lệnh open(QIODevice::WriteOnly | QIODevice::Text) nếu không mở được chúng ta phát tín hiệu tbloiTaoFile();

Cuối cùng là đóng file và thông báo hộp thoại với nội dung là “Tao file thang cong”.



Hình 51. Hàm thông báo lỗi tạo thư mục

Đoạn mã lệnh dưới đây là đoạn mã lệnh thêm giá trị port vào ListWidget ở trong tab2 mỗi khi thực hiện kết nối thành công với port hiện tại đang mở.



Hình 52. Hàm con trỏ trả về đối tượng ListWidgetItem



Hình 53. Hàm xoá, ghi kết quả và ghi IP vào file

**Hàm xoaDuLieuFile():** hàm này chúng ra sẽ mở con trỏ file ở chế độ Truncate chế độ này sẽ xoá toàn bộ dữ liệu trong file. Hàm này được gọi khi chúng ta cần reset chương trình và mỗi lần quét mới địa chỉ IP.

**Hàm ghiKetQuaRaFile():** trong hàm này chúng ta thực hiện tạo string ketQua và truyền vào giá trị port hiện tại sau mở file ở chế độ ghi liên tục, tạo một đối tượng out và đọc fileKetQua. Ghi vào file string mới tạo ketQua, sau cuối đóng file lại.

**Hàm ghiIP():** được thực hiện ghi địa chỉ IP hiện tại ra file và thời gian quét IP này quan thư viện Qtime với phương thức là currentTime().

### Hoàn thiện chương trình

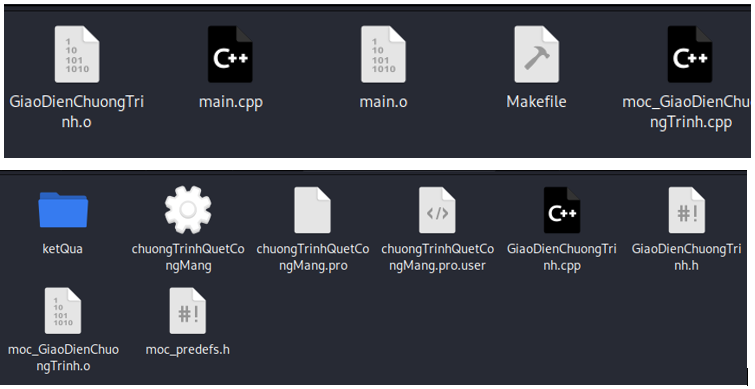
Hoàn thiện chương trình:



Hình 54. Hàm main của chương trình

Trong hàm main.cpp, chúng ta tạo ra một đối tượng giaoDien có kiểu class là GiaoDien. Sau đó gọi phương thức show().

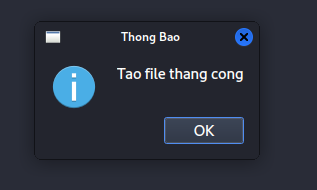
Kết quả khi chạy chương trình vào tạo chương trình:



Hình 55. kết quả của chương trình gồm các file

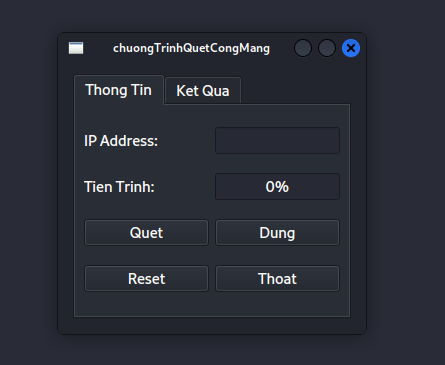
Chương trình tạo ra các file cài đặt từ 3 file: main.cpp, GiaoDienChuongTrinh.h và GiaoDienChuongTrinh.cpp khi qmake chương trình sẽ tạo ra file.pro, file này sẽ có tên chính bằng tên thư mục chứa nó là chuongTrinhQuetCongMang.pro.

Bắt đầu chạy chương trình chúng ta sẽ nhận được thông báo hộp thoại:



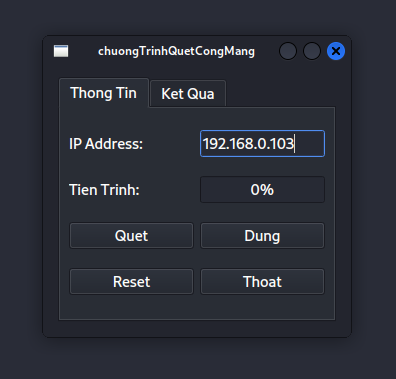
Hình 56. Hộp thoại tạo file thành công

Sau khi nhấn OK, chúng ta sẽ nhận được giao diện chương trình đã tạo:



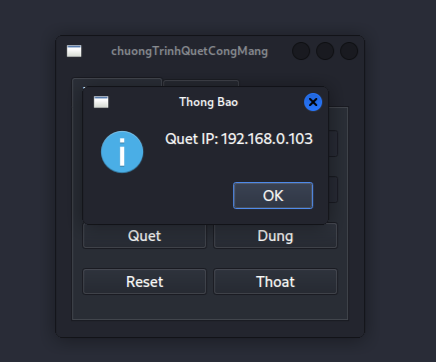
Hình 57. Giao diện chương trình

Tiếp tục nhập địa chỉ IP cần quét, ở đây chúng ta sẽ quét chính IP của máy mình:



Hình 58. Nhập IP cần quét

Sau khi nhập IP ở đây là 192.168.0.103 chúng ta bắt đầu nhấn nút Quet.

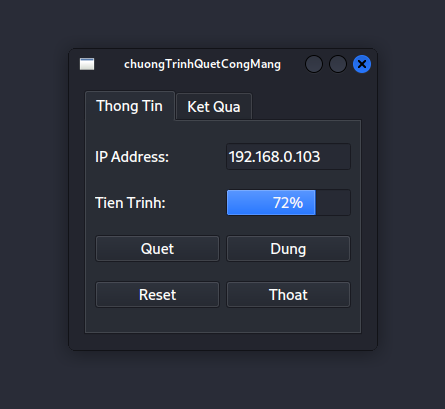


Hình 59. Thông báo quét IP hiện tại

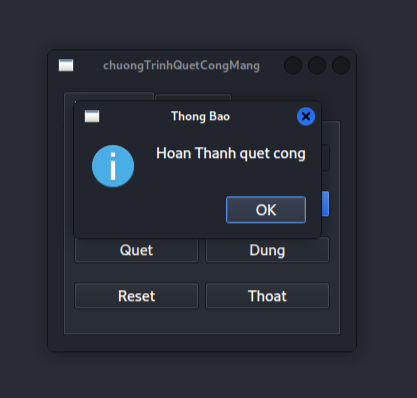
Hộp thoại thông báo quét IP hiện tại được gọi đến.



Hình 60. Tiến trình thay đổi

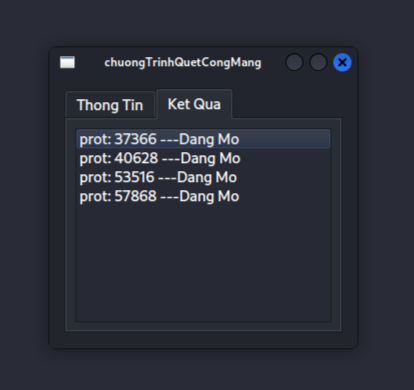


Hình 61. Tiến trình thay đổi



Hình 62. Thông báo hoàn thành quét cổng

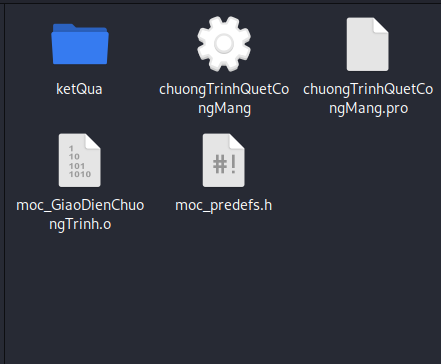
Sau khi quét cổng thành công hộp thoại thông báo Hoàn thành quét cổng được gọi đến. chúng ta sẽ chuyển sang tab2 là tab kết quả xem kết quả chúng ta thu được là những cổng nào đang mở:



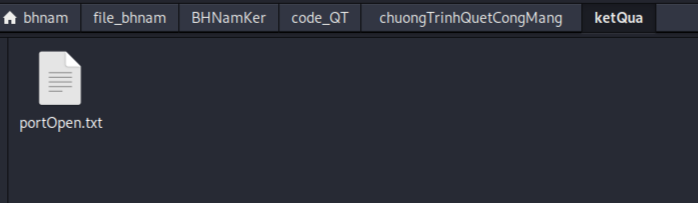
Hình 63. Kết quả các cổng mở được quét thấy

Kết quả thu được là các cổng 37366, 40628, 53316 và 57868 là có cổng đang nhận kết nối và phản hồi trong quá trình chúng ta quét cổng.

Kiểm tra thì ta nhận được một folder có tên là kết quả, click vào có một file là portOpen.txt.

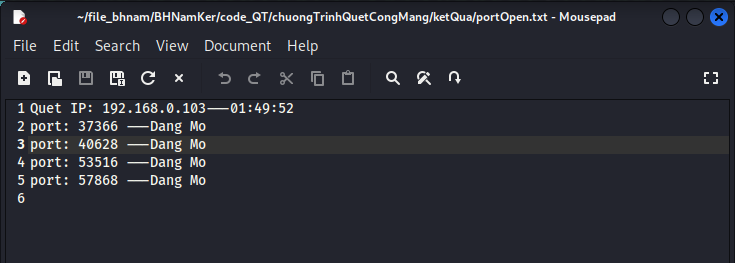


Hình 64. Đã tạo thư mục thành công



Hình 65. Tạo file thành công

Mở file portOpen.txt chúng ta nhận được kết quả:



Hình 66. Kết quả trên file

Vậy là tất cả các cổng mở đã được ghi ra file thành công.

## Đánh giá các nguy cơ đối với các cổng dịch vụ mạng

Việc quét cổng (port scanning) là một hoạt động quan trọng trong việc phát hiện lỗ hổng bảo mật trên các hệ thống máy tính. Khi chúng thực hiện quét cổng, chương trình quét sẽ kiểm tra các cổng mạng trên một máy tính hoặc một mạng để xác định xem những cổng nào đang mở và sẵn sàng để truy cập.

**Cổng từ 0 đến 1023**: cổng được gọi là "well-known ports", thường được sử dụng bởi các ứng dụng phổ biến như HTTP (port 80), FTP (port 21), SSH (port 22), Telnet (port 23), SMTP (port 25), DNS (port 53)… Các cổng này cũng thường được dùng cho các dịch vụ quan trọng của hệ thống.

**Cổng từ 1024 đến 49151**: cổng này được gọi là "registered ports", và thường được sử dụng bởi các ứng dụng được đăng ký trong IANA (Internet Assigned Numbers Authority), bao gồm các ứng dụng tùy chỉnh của các công ty hoặc tổ chức.

**Cổng từ 49152 đến 65535**: cổng này được gọi là "dynamic or private ports", và thường được sử dụng cho các kết nối tạm thời hoặc kết nối đến các ứng dụng không phải là của hệ thống. Các cổng này được sử dụng khi một máy tính truy cập vào một dịch vụ trên một máy tính khác mà không có thông tin về các cổng sẵn có.

Port 37366 và port 40628 ví dụ trong chương trình quét đều không được sử dụng phổ biến và thường không được liên kết với các ứng dụng hoặc dịch vụ cụ thể. Tuy nhiên, việc phát hiện các cổng này đang mở có thể cho thấy rằng máy tính hoặc mạng đang chạy một ứng dụng chưa biết hoặc có thể đang có các lỗ hổng bảo mật.

*Cụ thể,* **Cổng 37366** là một trong các cổng được sử dụng bởi ứng dụng RAT (Remote Access Trojan) có tên gọi DarkComet. Đây là một loại phần mềm độc hại có khả năng xâm nhập vào hệ thống từ xa và truy cập vào các thông tin nhạy cảm của người dùng, bao gồm thông tin tài khoản đăng nhập, các dữ liệu quan trọng và cả điều khiển hệ thống từ xa. Nếu cổng 37366 được sử dụng trái phép, nó có thể cho phép kẻ tấn công truy cập vào hệ thống và kiểm soát. Nguy cơ tấn công trên cổng 37366 là khi tên miền được đăng ký để giả mạo một trang web đáng tin cậy và phần mềm DarkComet được cài đặt trên trang web giả mạo. Khi người dùng truy cập trang web giả mạo, phần mềm DarkComet sẽ tự động tải xuống và cài đặt trên máy tính của nạn nhân. Sau đó, kẻ tấn công có thể tạo kết nối với máy tính của nạn nhân thông qua máy chủ điều khiển từ xa và thực hiện các hoạt động độc hại, bao gồm xem, sửa đổi hoặc xóa dữ liệu trên máy tính nạn nhân, và thậm chí kiểm soát máy tính từ xa.

**Cổng 40628** là một trong các cổng được sử dụng bởi các chương trình độc hại có khả năng tấn công vào các hệ thống thông tin định vị GPS (Global Positioning System). Nếu các kẻ tấn công sử dụng cổng này trái phép, họ có thể thực hiện các cuộc tấn công đối với các hệ thống GPS, bao gồm việc giả mạo các thông tin định vị và làm giảm chính xác của các thông tin định vị. Nguy cơ tấn công trên cổng 40628 là khi kẻ tấn công sử dụng một chương trình độc hại để tấn công vào hệ thống định vị GPS trên xe hơi. Khi chương trình độc hại được kích hoạt, nó sẽ bắt đầu gửi các gói tin độc hại tới thiết bị định vị GPS trên xe hơi. Những gói tin này có thể làm giảm độ chính xác của hệ thống, gây nhiễu sóng GPS, hay thậm chí làm hỏng toàn bộ hệ thống. Nếu cuộc tấn công thành công, người tấn công có thể kiểm soát hoàn toàn hệ thống định vị GPS trên xe hơi và sử dụng nó để theo dõi vị trí của xe hơi, hoặc thực hiện các hoạt động độc hại khác.

*Các cuộc tấn công bảo mật trên các cổng từ 0 đến 1023 thường liên quan đến các dịch vụ mạng phổ biến:*

* **Cuộc tấn công Denial-of-Service (DoS):** Kẻ tấn công sử dụng các công cụ hoặc kỹ thuật để làm gián đoạn hoạt động của dịch vụ đang chạy trên các cổng này. Điều này có thể dẫn đến việc làm gián đoạn hoạt động của hệ thống, làm cho hệ thống không thể truy cập được hoặc dẫn đến mất dữ liệu quan trọng.
* **Cuộc tấn công Buffer Overflow:** Kẻ tấn công sử dụng các lỗ hổng bảo mật trong các dịch vụ mạng để gửi các yêu cầu lớn hơn kích thước đệm được cấp phát. Điều này có thể dẫn đến việc ghi đè lên bộ nhớ và kiểm soát hệ thống, cho phép kẻ tấn công thực hiện các hoạt động không được ủy quyền.
* **Cuộc tấn công Man-in-the-middle (MITM):** Kẻ tấn công sử dụng các kỹ thuật để thay đổi hoặc đánh cắp thông tin giao tiếp giữa hai bên đang trao đổi thông tin trên mạng. Điều này có thể dẫn đến việc lộ thông tin quan trọng như tài khoản ngân hàng, mật khẩu, thông tin cá nhân,...

*Dưới đây là một số ví dụ về các cuộc tấn công bảo mật trên cổng 1024 đến 49151:*

* **Tấn công Syn Flood:** Tấn công Syn Flood thường nhắm vào cổng 80, 443 và các cổng HTTP/HTTPS khác. Tấn công này sử dụng các yêu cầu kết nối TCP giả mạo để tạo ra các kết nối TCP đang chờ xử lý trên máy chủ mục tiêu. Khi số lượng kết nối chờ xử lý trên máy chủ vượt quá giới hạn cho phép, máy chủ sẽ không thể xử lý các yêu cầu mới, dẫn đến tình trạng tấn công từ chối dịch vụ (DoS).
* **Tấn công Brute Force:** Tấn công Brute Force là một phương pháp tấn công bằng cách thử tất cả các khả năng mật khẩu để đăng nhập vào hệ thống. Điều này thường được thực hiện trên các ứng dụng web hoặc các giao thức đăng nhập từ xa như SSH hoặc Telnet, thường sử dụng cổng 22 và 23.
* **Tấn công DNS Spoofing:** Tấn công DNS Spoofing có thể được thực hiện trên bất kỳ cổng nào sử dụng giao thức DNS. Tấn công này đánh lừa máy tính mục tiêu bằng cách gửi các phản hồi DNS giả mạo, dẫn đến việc máy tính không thể truy cập được các trang web hoặc được chuyển hướng đến các trang web giả mạo.
* **Tấn công SQL Injection:** Tấn công SQL Injection thường được thực hiện trên các cổng sử dụng giao thức HTTP, đặc biệt là các ứng dụng web chạy trên các máy chủ web như Apache hoặc IIS. Kẻ tấn công có thể chèn các lệnh SQL độc hại vào các trang web để thực hiện các cuộc tấn công khác hoặc đánh cắp thông tin từ cơ sở dữ liệu.
* **Tấn công Remote Code Execution:** Tấn công Remote Code Execution thường được thực hiện trên các cổng sử dụng giao thức HTTP, đặc biệt là các ứng dụng web chạy trên các máy chủ web như Apache hoặc IIS. Kẻ tấn công có thể tải lên mã độc độc hại hoặc thực thi các lệnh từ xa trên máy chủ mục tiêu.

*Dưới đây là một số ví dụ cụ thể về các cuộc tấn công bảo mật đã sử dụng các cổng động có số port từ 49152 đến 65535:*

* **Cuộc tấn công mã độc WannaCry:** Cuộc tấn công WannaCry là một cuộc tấn công phát tán phần mềm độc hại ransomware vào năm 2017. Mã độc này đã khai thác lỗ hổng bảo mật EternalBlue trên các hệ thống Windows, mà phổ biến trong các máy tính đang chạy Windows XP đến Windows 7. Mã độc WannaCry đã sử dụng cổng 445 và các cổng động có số port từ 49152 đến 65535 để lây nhiễm và lan truyền trên các hệ thống mạng.
* **Cuộc tấn công DDoS Mirai:** Cuộc tấn công DDoS Mirai vào năm 2016 đã tấn công các thiết bị IoT (Internet of Things) như máy quạt thông minh, máy in, camera an ninh, v.v. để tạo ra một mạng botnet và thực hiện cuộc tấn công từ chối dịch vụ trên các trang web. Mirai đã sử dụng các cổng động có số port từ 49152 đến 65535 để tìm kiếm các thiết bị IoT trên mạng và lây nhiễm chúng.
* **Cuộc tấn công Ngày Đen 25 tháng 5 (BlackNurse):** BlackNurse là một cuộc tấn công từ chối dịch vụ trên mạng (DoS) phát hiện vào năm 2016. Cuộc tấn công này đã sử dụng các gói tin ICMP (Internet Control Message Protocol) với kích thước nhỏ để tấn công các thiết bị mạng, làm cho chúng bị quá tải và từ chối dịch vụ. BlackNurse đã sử dụng các cổng động có số port từ 49152 đến 65535 để thực hiện cuộc tấn công này.

Việc đóng các cổng không được sử dụng là một phương pháp cơ bản để cải thiện bảo mật hệ thống. Nếu không có bất kỳ ứng dụng hoặc dịch vụ nào được sử dụng trên các cổng này, có thể cân nhắc đóng các cổng này để ngăn chặn các tấn công từ bên ngoài. Tuy nhiên, trước khi đóng các cổng này, nên kiểm tra lại để đảm bảo rằng việc đóng cổng không ảnh hưởng đến hoạt động của các ứng dụng hoặc dịch vụ khác trên hệ thống.

Đóng cổng trên Linux: chúng ta sử dụng, UFW (Uncomplicated Firewall) là một công cụ quản lý tường lửa trên hệ điều hành Linux. Nó là một giao diện đơn giản cho iptables, cho phép người dùng quản lý quy tắc tường lửa một cách dễ dàng.

Mở terminal trên hệ thống.

Gõ lệnh sau**: sudo ufw deny <port number>/tcp** hoặc **sudo ufw deny <port number>/udp**

Nhập mật khẩu của người dùng quản trị và nhấn Enter để thực thi lệnh.

Lưu ý rằng việc đóng các cổng không được sử dụng có thể gây ảnh hưởng đến các ứng dụng hoặc dịch vụ khác trên hệ thống của bạn, vì vậy hãy đảm bảo rằng bạn đã kiểm tra kỹ trước khi thực hiện hành động này.

Quét cổng (port scanning) là một hoạt động có nguy cơ và có thể gây ra nhiều rủi ro đối với hệ thống. Sau đây là một số rủi ro tiêu biểu mà quét cổng có thể gây ra:

* **Tấn công từ chối dịch vụ (DoS):** Nếu kẻ tấn công quét cổng và tìm ra các dịch vụ đang chạy trên hệ thống, họ có thể sử dụng thông tin đó để tấn công các dịch vụ đó bằng cách gửi lượng lớn yêu cầu đến hệ thống, khiến cho các dịch vụ không thể hoạt động đúng cách hoặc ngừng hoạt động hoàn toàn. Điều này được gọi là tấn công từ chối dịch vụ (DoS). Hoạt động trên hầu hết các cổng dịch vụ.
* **Lây nhiễm phần mềm độc hại**: Kẻ tấn công có thể quét cổng để tìm kiếm các lỗ hổng bảo mật trên hệ thống của bạn và sử dụng chúng để lây nhiễm phần mềm độc hại vào hệ thống. thường hoạt động trên các cổng: 135 (Microsoft EPMAP service), cổng 139 (NetBIOS Session Service), cổng 445 (Microsoft-DS SMB file sharing), cổng 3389 (Remote Desktop Protocol - RDP), cổng 5900 (Virtual Network Computing - VNC), cổng 8080 (HTTP alternate)
* **Xâm nhập trái phép:** Kẻ tấn công có thể quét cổng để tìm kiếm các lỗ hổng bảo mật trên hệ thống và sử dụng chúng để truy cập trái phép vào hệ thống. Hoạt động trên nhiều cổng khác nhau và tùy thuộc vào mục tiêu và kỹ năng của kẻ tấn công, Các cổng thường được sử dụng cho các dịch vụ đặc quyền như cổng 22 (SSH), cổng 3389 (Remote Desktop Protocol), cổng 5900 (Virtual Network Computing), cổng 1433 (Microsoft SQL Server), cổng 3306 (MySQL) và cổng 1521 (Oracle Database). Các cổng được sử dụng cho các dịch vụ phổ biến như cổng 80 (HTTP), cổng 443 (HTTPS), cổng 25 (SMTP), cổng 110 (POP3), cổng 143 (IMAP) và cổng 53 (DNS). Các cổng trống được sử dụng để tìm kiếm các hệ thống chưa được bảo vệ, cho phép kẻ tấn công thực hiện các cuộc tấn công quét mạng và xác định các hệ thống đích để tấn công.
* **Làm lộ thông tin:** Kẻ tấn công có thể quét cổng để tìm kiếm các cổng mở trên hệ thống và tìm ra các dịch vụ đang chạy. Thông tin này có thể được sử dụng để đánh giá các lỗ hổng bảo mật trong hệ thống của bạn và lợi dụng để truy cập trái phép vào hệ thống. Cổng 80 (HTTP) khi truy cập trang web không được bảo mật, thông tin truy cập có thể bị lộ. Cổng 443 (HTTPS) mặc dù sử dụng giao thức bảo mật SSL/TLS, tuy nhiên nếu chứng chỉ không hợp lệ hoặc không được cài đặt đúng cách, thông tin vẫn có thể bị lộ. Cổng 21 (FTP) khi sử dụng giao thức FTP không được bảo mật, thông tin đăng nhập có thể bị lộ. Cổng 25 (SMTP) khi sử dụng giao thức SMTP để gửi email không được bảo mật, nội dung email và thông tin người gửi, người nhận có thể bị lộ. Cổng 143 (IMAP) và cổng 110 (POP3) khi sử dụng giao thức email để lấy email không được bảo mật, nội dung email và thông tin người gửi, người nhận có thể bị lộ.

Do đó, để bảo vệ hệ thống khỏi các cuộc tấn công quét cổng, cần thực hiện các biện pháp bảo mật như đóng cổng không sử dụng, cập nhật phần mềm đầy đủ và định kỳ, cài đặt và cấu hình tường lửa, và kiểm tra các thiết lập bảo mật trong hệ thống.

**TÀI LIỆU THAM KHẢO**

1. Phillip A. Laplante (2001). Dictionary of Computer Science, Engineering, andTechnology. CRC Press. tr. 94–95.
2. John L. Hennessy and David A. Patterson (2003). Computer Architecture: A Quantitative Approach.
3. Nguyễn Phú Trường, Cấu trúc hệ điều hành, Đại học Cần Thơ - Khoa công nghệ thông tin - Giáo trình hệ điều hành.
4. Tổng quan về hệ điều hành Windows: https://bogounvlang.org
5. Kiến trúc của hệ điều hành linux: https://kb.pavietnam.vn
6. Lịch sử Linux: https://vi.wikipedia.org
7. Unix: https://vi.wikipedia.org
8. Internet: https://vi.wikipedia.org
9. Các dịch vụ Interner được tìm kiếm và phổ biến: http://thietkeweb180.com
10. TCP/IP: https://vi.wikipedia.org
11. F. Kurose 2013, tr. 52, "1.5.1 Layered Architecture Link Layer" & Internet Engineering Task Force 1989, tr. 10, 1.1.3 Internet Protocol Suite "Link layer".
12. http://www.laptrinhtanbinh.com

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

Thái Nguyên, ngày tháng năm 2023

GIÁO VIÊN HƯỚNG DẪN