Câu 1: Định nghĩa hệ thống nhúng

Câu 2: Các thành phần chính của hệ thống nhúng

Câu 3: Đặc điểm của hệ thống nhúng

Câu 4: Tại sao nói hầu hết các hệ thống nhúng hoạt động với ràng buộc về thời gian

Câu 5: Các yêu cầu đối với hệ thống nhúng

Câu 6:Quá trình thiết kế hệ thống nhúng

Câu 7: Hệ thống nhúng tương tác với môi trường vật lý như thế nào

Câu 8: Các tiêu chí phân loại hệ thống nhúng

Câu 9: Các kiểu hoạt động của hệ thống nhúng

Câu 1: **Tổng quan phần cứng của Hệ thống nhúng**

**Câu 2: Thành phần cơ bản của CPU**

**Câu 2.1: Phân loại CPU**

**Câu 3: Quá trình thực thi 1 lệnh**

**Câu 4: RISC và CISC**

**Câu 5: Von Neumann và Harvard**

**Câu 6:SRAM và DRAM**

**Câu 7: Ghép nối với thiết bị ngoại vi**

**Câu 8: 8051**

Câu 1: Device Driver (Trình điều khiển thiết bị)

Câu 2: Real Time Operating System (RTOS)

Câu 3: Middleware

Câu 4: Phát triển phần mềm nhúng cần những gì

Câu 5 Các bước biên dịch phần mềm trong IDE

Chương I.

Câu 1: Định nghĩa hệ thống nhúng

Hệ thông nhúng là một thuật ngữ để chỉ những hệ thống có khả năng tự trị được nhúng vào trong một môi trường hay một hệ thống khác có quy mô phức tạp hơn. Đó là những hệ thống có phần cứng ( là những máy tính được xây dụng trên cơ sở vi xử lý) và phần mềm nhúng được nhúng vào phần cứng để thực hiện các nhiệm vụ chức năng chuyên biệt

Câu 2: Các thành phần chính của hệ thống nhúng

* Phần cứng: Processer, Memmory(RAM/ROM/EPRAM/Flash), I/0, nguồn cung cáp, Timer/counter,...
* Phần mềm: Devide Driver, Hệ điều hành thời gian thực (RTOS), phần mềm trung gian(middle ware), phần mềm ứng dụng

Câu 3:Đặc điểm của hệ thống nhúng

* Thực hiện một hoặc 1 số chức năng chuyên biệt
* Hệ điều hành thời gian thực (RTOS)
* Tính tin cậy, khả năng hoạt động ở các môi trường khắc nghiệt
* Giá thành hợp lý, thường nhỏ gọn, có thể di chuyển dễ dàng giữa các địa điểm
* Có tài nguyên hạn chế, sử dụng ít năng lượng
* Tính an toàn và bảo mật
* Tương tác được với các môi trường bên ngoài qua các cảm biến và DAC ADC

Câu 4: Tại sao nói hầu hết các hệ thống nhúng hoạt động với ràng buộc về thời gian

Các hệ thống nhúng hoạt động ràng buộc về thời gian vì chúng thường thực hiện các nhiệm vụ trong thời gian thực hoặc gần thời gian thực.

* Yêu cầu về thời gian thực: Các hệ thống nhúng như trong y tế, hàng không, vũ trụ có những yêu cầu rất sát sao về độ lệch thời gian. Một sai xót nhỏ có thể gây ra hậu quả rất lớn
* Thông thường các hệ thống nhúng có tài nguyên hạn chế => để tránh không lãng phí tài nguyên nên rất nghiêm ngặt về mặt thời gian
* Nhu cầu người dùng
* Liên kết với các thiết bị và hệ thống khác

Câu 5: Các yêu cầu đối với hệ thống nhúng

1. Xử lý các đầu vào với thời gian thực hoặc gần thời gian thực
2. Có thể hoạt động tốt ở môi trường khắc nghiệt
3. Sử dụng tiết kiệm năng lương, có thể sử dụng pin và ắc quy để hoạt động
4. Tốc đô xử lý cao, hoạt động tin cậy, chịu lỗi cao
5. Giá thành hợp lý
6. Gọn nhẹ dễ dàng di chuyển
7. An toàn và bảo mật thông tin
8. Khả năng nâng cấp phần mềm và dự phòng nâng cấp phần cứng

Câu 6:Quá trình thiết kế hệ thống nhúng

1. Xác định đối tượng/yêu cầu của hệ thống
2. Xác định cấu trúc của phần cứng/phần mềm
3. Các chức năng bổ sung
4. Tìm hiểu các họ cùng thiết kế liên quan
5. Chia thiết kế thành các module nhỏ(phần cứng, phần mềm)
6. Mapping
7. Thiết kế giao diện hệ thống
8. Hiệu chỉnh hệ thống

Câu 7: Hệ thống nhúng tương tác với môi trường vật lý như thế nào

* Hệ thống nhúng tương tác với môi trường vật lý bên ngoài thông qua biến đổi các tín hiệu thu được từ các thiết bị I/O và cảm biến
  + DAC
  + ADC
  + Phương tiện để tương tác là ghép nối và hợp chuẩn dữ liệu

Câu 8: Các tiêu chí phân loại hệ thống nhúng

* Hê thống nhúng hoạt động ở đâu (độc lâp/ có liên kết)
* Lĩnh vực (Y tế, sản xuất, hàng không, vũ trụ,..)
* Quy mô (nhỏ, vừa , lớn)

Chương II.

Câu 1: Tổng quan về phần cứng của Hệ thống nhúng

* Vi xử lý CPU: thực hiện lệnh và xử lý dữ liệu
* Memory : RAM/ROM/EPROM/Flash
* Hệ thống BUS
* Nguồn cung cấp
* Thiết bị ngoại vi I/O, cảm biến
* Counter/Timer

Câu 2: Thành phần cơ bản của CPU

* CU: Đơn vị điều khiển
* ALU: đơn vị tính toán
* Memmory: RAM/ROM/Cache L1/Register
* Hệ thống BUS(data/adress/control)

Câu 3: Phân loại CPU

* Theo kiến trúc tập lệnh
  + RISC
  + CISC
* Theo kiểu kiến trúc
  + Von neumann
  + Havard
* Công nghệ ché tạo hướng ứng dụng
  + CPU đa năng
  + CPU chuyên dụng

Câu 4: Quá trình thực thi 1 lệnh

1. Lệnh được nạp từ bộ nhớ vào CU
2. CU giải mã lệnh
3. ALU thực thi lệnh
4. Kết quả được lưu vào bộ nhớ

Câu 5: RISC và CISC

* RISC: ít lệnh, lệnh đơn giản, chương trình dài, hỗ trợ pipeline, hiệu suất cao
* CISC: nhiều lệnh, lệnh phức tạp, chương trình ngắn, ít hỗ trợ pipeline, hiệu suất thấp hơn

Câu 6: Von Neumann và Harvard

* Von Neumann
  + Đặc điểm
    - sử dụng một bộ nhớ duy nhất để lưu trữ cả chương trình (các tập lệnh) và dữ liệu.
    - sử dụng cùng một bus cho cả dữ liệu và chương trình
    - Chương trình được thực hiện tuần tự theo thứ tự các lệnh được lưu trữ trong bộ nhớ
* Harvard
  + Đặc điểm
    - có hai không gian bộ nhớ vật lý tách biệt: một cho lệnh (chương trình) và một cho dữ liệu
    - Sử dụng các bus riêng biệt để truyền tải lệnh và dữ liệu
    - Chương trình có thể được thực hiện song song để tăng hiệu suất hệ thống

Câu 7: SRAM và DRAM

* SRAM thường là cache
* DRAM là bộ nhớ chính

Câu 8: Ghép nối với thiết bị ngoại vi

* Ghép nối CPU chủ động
  + Có điều kiện
  + Không điều kiện (đối thoại)
* Ghép nối I/O chủ động
  + Ngắt(Interrupt): Khi có yêu cầu từ thiết bị ngoại vi đên CPU, CPU sẽ tạm dựng chương trình hiện tại và cho nó vào ngăn xêp stack để thực hiện lệnh được yêu cầu. Khi thực hiện xong thì tiếp tục thực hiện chương trình trong stack
  + Truy cập trực tiếp bộ nhớ (DMA): Thiết bị ngoại vị sẽ truy cập trực tiếp vào bộ nhớ mà không cần thông qua CPU,CPU sẽ ra lệnh cấp quyền cho DMAC, ngoại vi sẽ thông qua DMAC (DMA controller) để lấy dữ liệu từ memmory.
* Hỗn hợp

Câu 9: Vi điều khiển 8051

* Kiến trúc cơ bản vdk 8051: vi xử lý, thanh ghi, bộ nhớ (tương tự kiến trúc Harvard) và các cổng, các thiết bị I/O, counter/timer và bộ xử lý ngắt
* Đặc điểm:
  + Xung nhịp vxl 12 MHz, chu kỳ lệnh 1 us
  + Kiến trúc Harvard
  + ALU 8 bit
  + Bus dữ liệu 8 bit, bus địa chỉ 16 bit
  + Không xử lý dấu phẩy động
  + Không có bộ nhớ Cache
  + Không có bộ quản lý bộ nhớ MMU
  + Không có CPU pipeline
  + Kích thước RAM 128 bytes
  + Không xử lý các lệnh song song
  + Kiến trúc tập lệnh CISC

Chương 3:

Câu 1: Device Driver (Trình điều khiển thiết bị)

* Kết nối giữa OS và các thiết bị I/O
* Chức năng: dịch, convert các yêu cầu từ OS thành các câu lệnh mà bộ điều khiển ngoại vi có thể hiểu được.
* Là một phần của nhân OS (drivers là các module phần mềm được xây dựng riêng biệt và cài vào OS khi cần thiết)
* Một hệ thống hoàn chỉnh có rất nhiều driver được cài đặt trên hệ điều hành
* Công dụng
  + Đơn giản hóa hoạt động của OS
  + Không có divice driver, OS chịu trách nhiệm giao tiếp trực tiếp với phần cứng => gây quá tải cho OS
  + Nếu một thiết bị mới được cài đặt, dẫn đến phải thay đổi OS
* Đặc điểm:
  + Driver là tập các entry points được gọi bởi OS
  + Một Driver bao gồm:
    - Cấu trúc data riêng biệt cho từng thiết bị
    - Giao thức riêng biệt cho từng driver
  + Hầu hết các drivers được viết trên 1 file nguồn
  + Phần đầu của driver được gọi là khi báo prologue
  + Khai báo (prologue) gồm:
    - #include
    - #define
    - Khai báo các biến và đặc điểm các biến
  + Phần còn lại của driver
    - Entry points (các hàm C được tham chiếu bởi OS)
    - Routines (các hàm C tương tác với driver)
* Phân loại:
  + Block Driver
    - Chức năng: Điều khiển thiết bị lưu trữ dưới dạng các block dữ liệu (khối)
    - Ví dụ: Driver cho ổ đĩa cứng, SSD, USB
    - Đặc điểm: Thường cho phép truy cập dữ liệu bất kỳ trong các khối một cách tùy ý
  + Character Driver
    - Chức năng: Điều khiển thiết bị lưu trữ dữ liệu dưới dạng ký tự hoặc byte riêng lẻ
    - Ví dụ: Thiết bị âm thanh, thiết bị nhập liệu như bàn phím hoặc chuột
    - Đặc điểm: Làm việc với dữ liệu một cách tuần tự và không có khái niệm khối
  + Terminal Driver
    - Chức năng: Điều khiển các thiết bị giao tiếp trực tuyến với người dùng qua giao diện dòng lệnh hoặc terminal
    - Ví dụ: Driver cho các thiết bị teletypewriter, console
    - Đặc điểm: Xử lý các tác vụ như nhập xuất dữ liệu, quản lý phiên bản làm việc (session), và kiểm soát các tính năng như cuộn màn hình.
  + Stream Driver
    - Chức năng: Điều khiển các thiết bị hoạt động dựa trên dòng dữ liệu liên tục (stream), thường được sử dụng trong việc truyền dữ liệu qua mạng hoặc các giao thức truyền tải dữ liệu như TCP/IP
    - Ví dụ: Driver cho card mạng Ethernet, giao diện loopback
    - Đặc điểm: Xử lý các dòng dữ liệu trực tiếp từ nguồn đến đích mà không cần lưu trữ dữ liệu tạm thời.

Câu 2:

* Hệ điều hành thời gian thực(RTOS) là hệ điều hành đa tác vụ cho các ứng dụng có ràng buộc (contrains) về thời gian thực, dùng trong lĩnh vực thiết kế hệ thống nhúng
* Ràng buộc về thời gian thực: yêu cầu một độ trễ xác định cho một sự kiện và đáp ứng của hệ thống
* Một thuộc tính quan trọng của RTOS là tính tách biệt với ứng dụng. Khi một chương trình bị “chết” hoặc hoạt động không hợp lệ, thì RTOS có thể thực hiện các điều sau
  + Cô lập chương trình này
  + Kích hoạt cơ chế phục hồi
  + Bảo vệ các chương trình khác
* Chức năng của RTOS là giải quyết, điều phối các tác vụ, lập lịch và phân cấp mức độ yêu tiên, tất cả điều trên hoạt động được là nhờ cơ chế:
  + Hướng sự kiện (Event driven): việc giải quyết, điều phối phụ thuộc vào mức độ ưu tiên của các tác vụ
  + Chia sẻ thời gian (Time sharing): Các tác vụ được chuyển giao nhờ vào cơ chế ngắt
* Các nhiệm vụ chính của RTOS
  + Xử lý ngắt
  + Dịch vụ quản lý thời gian
  + Dịch vụ quản lý thiết bị
  + Dịch vụ quản lý bộ nhớ
  + Dịch vụ quản lý vào ra
  + Lập lịch
  + Đồng bộ và thông điệp
* Phân loại RTOS
  + Hard RTOS: yêu cầu về thời gian thực giải quyết các tác vụ phải diễn ra trong mọi trường hợp
  + Soft RTOS: có yêu cầu về thời gian thực trong giải quyết các tác vụ tuy nhiên một số trường hợp có thể bị trễ, trong trường hợp này phải có các cơ chế bù trừ
* OS và RTOS

|  |  |  |
| --- | --- | --- |
|  | OS | RTOS |
| Ràng buộc về thời gian xử lý | Linh hoạt cho thời gian xử lý các tác vụ, trễ tự do | Thời gian xử lý cố định, cứng nhắc, trễ xác định |
| Quản lý tài nguyên | Tài nguyên được chia linh hoạt, có thể chia sẻ cho nhau | Tài nguyên cố đinh, mỗi tác vụ được cấp tài nguyên nghiêm ngặt |
| Các ứng dụng | Windows, MacOS,Linux | FreeRTOS,QNX,.. |

Câu 3: Middleware

* Nằm ở nhiều nơi, tách biệt với app, OS, driver
* Khai niệm:
  + Coi như phần mềm trung gian, giúp đảm bảo tính linh hoạt, an ninh, giúp liên kết với các phần mềm khác ở những lớp kề cận
  + Gồm các hàm dịch vụ, được gọi và sử dụng nhiều lần bởi các phần mềm khác

Câu 4: Phát triển phần mềm nhúng cần những gì

1. Phần mềm biên dịch
2. Board phát triển
3. Bộ nạp/ Bộ gỡ lỗi
4. Thư viện (cho lõi ARM, driver ngoại vi,...)
5. Tài liệu (datasheet)

Câu 5: Các bước biên dịch phần mềm trong IDE

1. Tạo Project
2. Cấu hình phần mềm biên dịch (chọn loại chíp, chọn loại mạch nạp)
3. Viết chương trình và biên dịch
4. Nạp vào bộ nhớ Flash của board phát triển
5. Chạy chương trình và kiểm tra lỗi

Câu 6: Các kiểu hoạt động của hệ thống nhúng

* Hệ thông minh
* Hệ hoạt động độc lập
* Hệ tự phản ứng với sự kiện
* Hệ liên kết tự động

Câu 7:

Bus của CPU gôm các thành phần hợp thành

* Bus dữ liệu
* Bus địa chỉ
* Bus điều khiển

Sự khác nhau giữa Bus của CPU và Bus hệ thống

* Bus CPU
  + Đi ra trực tiếp từ CPU
  + Bus dồn kênh
* Bus hệ thống
  + Không nối trực tiếp vào CPU mà qua khuếch đại bus
  + Không dồn kênh, các bus tách biệt
  + Có bus clock