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 1

**Câu1: Định nghĩa về Hệ thống nhúng**

* Hệ thống nhúng là một thuật ngữ để chỉ một hệ thống có khả năng hoạt động tự trị được nhúng vào trong một môi trường hay một hệ thống khác có quy mô phức tạp hơn. Đó là các hệ thống tích hợp cả phần cứng (là một hệ thống máy tính được xây dựng trên cơ sở vi xử lý) và phần mềm nhúng trong phần cứng đó, để thực hiện các bài toán và tác vụ chuyên biệt
* Theo IEEE: Hệ thống nhúng là một hệ tính toán (máy tính số) nằm trong (hay được nhúng vào) sản phẩm khác lớn hơn và thường ẩn đối với người sử dụng. Nói rộng ra, và đơn giản hơn, khi một hệ tính toán được nhúng vào trong một sản phẩm hay một hệ thống nào đó và thực hiện một số chức năng cụ thể của hệ thống đó, thì ta gọi hệ tính toán đó là một hệ thống nhúng.

**Câu 2: Các thành phần chính của hệ thống nhúng**

1. Phần cứng (processor , ROM, RAM,I/O, Power supply,....)
2. Phần mềm nhúng
3. Real time Operating System (RTOS)

**Câu 3: Đặc điểm của hệ thống nhúng**

* Hàm chức năng chuyên dụng
* Real time operations
* Có độ tin cậy cao và hoạt động trong các môi trường khắc nghiệt
* Tương tác được với môi trường bên ngoài qua nhiều phương thức
* Có hiệu năng cao, trọng lượng thường nhỏ gọn, giá thành hợp lý.

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

Hầu hết các hệ thống nhúng hoạt động với sự ràng buộc về thời gian vì chúng thường thực hiện các nhiệm vụ trong thời gian thực hoặc gần thời gian thực. Điều này có nghĩa là chúng phải đáp ứng các yêu cầu về thời gian nghiêm ngặt để đảm bảo hoạt động chính xác và hiệu quả. Các lý do chính bao gồm:

* **Yêu cầu thời gian thực**: Nhiều hệ thống nhúng được thiết kế để điều khiển và quản lý các quá trình vật lý, như hệ thống phanh ABS trong xe hơi, máy điều hòa không khí, hoặc robot công nghiệp. Các hệ thống này phải phản hồi trong một khoảng thời gian rất ngắn để đảm bảo an toàn và hiệu suất.
* **Độ tin cậy và tính chính xác**: Trong các ứng dụng như thiết bị y tế, hệ thống điều khiển không gian, và thiết bị hàng không, việc không đáp ứng đúng hạn thời gian có thể dẫn đến hậu quả nghiêm trọng. Do đó, hệ thống phải được thiết kế để đảm bảo thời gian đáp ứng chính xác.
* **Tối ưu hóa tài nguyên**: Hệ thống nhúng thường có tài nguyên hạn chế như bộ nhớ, bộ vi xử lý và năng lượng. Để tối ưu hóa hiệu suất, các hệ thống này cần quản lý thời gian thực thi các nhiệm vụ một cách chặt chẽ, đảm bảo không có tài nguyên nào bị lãng phí.
* **Đáp ứng nhu cầu người dùng**: Trong các thiết bị tiêu dùng như điện thoại thông minh, TV thông minh và các thiết bị gia dụng khác, người dùng mong đợi sự phản hồi nhanh chóng và mượt mà. Điều này đòi hỏi hệ thống nhúng phải xử lý các yêu cầu người dùng trong thời gian thực hoặc gần thời gian thực.
* **Khả năng tương tác và điều khiển**: Các hệ thống nhúng thường cần tương tác với các thiết bị hoặc hệ thống khác trong thời gian thực, chẳng hạn như trong mạng cảm biến không dây hoặc hệ thống tự động hóa. Sự chậm trễ trong việc xử lý dữ liệu có thể gây ra lỗi hoặc làm giảm hiệu suất của toàn bộ hệ thống.

**Câu 5: Các yêu cầu với hệ thống nhúng**

* Khả năng đáp ứng với sự kiện bên ngoài (từ các tác nhân bị kiểm soát) phải nhanh nhạy, kịp thời, tức là khả năng theo thời gian thực:
* Có khả năng làm ở môi trường khắc nghiệt.
* Có giá thành thấp hay hiệu quả hoạt động/giá thành hợp lí.
* Kích thước nhỏ gọn, nhẹ, dễ mang dễ vận chuyển, lắp đặt.
* Tiêu thụ năng lượng thấp, khả năng sử dụng nguồn pin, ắc qui (tất nhiên phụ thuộc vào dung lượng của pin, ắc qui).
* Hoạt động tin cậy, chịu lỗi cao
* An toàn và bảo mật.
* 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 hệ thống
2. Xác định cấu trúc phần cứng, phần mềm
3. Các chức năng bổ sung
4. Tìm hiểu các họ cùng thiết kế liên quan
5. Chia thiết kế ra 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 bằng các chuyển đổi các tín hiệu thu được qua các thiết bị I/O

* Biến đổi tương tự thành số ADC
* Biến đổi số thành tương tự DAC
* Phương tiên để tương tác là ghép nối và hợp chuẩn tín hiệu

**Câu 8: Các tiêu chí phân loại hệ thống nhúng**

* Hệ thống nhúng hoạt động ở đâu (độc lập, có liên kết)
* Lĩnh vực(tính toán, xử lý tín hiệu, truyền thồn, điều khiển,...)
* Kiến trúc và quy mô

Chương 2:

Thiếu DMA(Direct memory access): phần cứng truy cập vào bộ nhớ mà không cần qua CPU (DMAC - DMA controler)

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

* + **CPU** (Bộ xử lý trung tâm): Thực hiện lệnh và xử lý dữ liệu
  + **Memory**: Bộ nhớ chương trình và dữ liệu
  + **Ngoại vi**: Input / output devices
  + **Bus**: truyền tín hiệu giữa các thành phần

**Câu 2: Thành phần cơ bản của CPU**

* CU: Control Unit: đơn vị điều khiển
* ALU: Đơn vị tính toán và logic
* Memory Unit: (RAM/ROM/Cache 1,Register...)
* Bus (data/adress/control)

**Câu 2.1: Phân loại CPU**

* Cách tổ chức và thực hiện lệnh máy: -
  + Với tập lệnh đầy đủ (CICS). -
  + Với tập lệnh rút gọn (RISC).
* Cách xử lý thông tin và truy nhập bộ nhớ:
  + Von Neumman: bộ nhớ chung, truy cập tuần tự theo từng lệnh máy.
  + Harvard: bộ nhớ lệnh và bộ nhớ dữ liệu độc lập, truy cập đồng thời.
* Công nghệ chế tạo hướng ứng dụng:
  + CPU đa năng: chế tạo máy tính đa năng, HTN tinh vi, hệ điều khiển thông minh.
  + CPU chuyên dụng: các ứng dụng đặc thù ( như cho ứng dụng nhúng)

**Câu 3: Quá trình thực thi 1 lệnh**

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

**Câu 4: RISC và CISC**



**Câu 5: Von Neumann và Harvard**

Von Neumann

Đặc điểm:  
- Cả chương trình và dữ liệu được lưu trữ trong cùng một không gian bộ nhớ.

- Chỉ có một bus duy nhất để truyền cả lệnh và dữ liệu.

Harvard

Đặc điểm:

- Mã chương trình và dữ liệu được lưu trữ trong các không gian bộ nhớ riêng biệt.

- Có bus riêng biệt cho lệnh và dữ liệu.

**Câu 6:SRAM và DRAM**

* SRAM thường là cache
* DRAM bộ nhớ chính

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

* CPU chủ động
  + Có điều kiện
  + Không điều kiện( đối thoại)
* I/O chủ động
  + Ngắt (Interrupt)
  + Truy cập trực tiếp bộ nhớ (DMA)
* Hỗn hợp

**Câu 8: 8051 (ÔN SAU)**

* Kiến trúc cơ bản 8051: vi xử lý, thanh ghi, bộ nhớ (trong kiến trúc Harvard) và các cổng, Counter/timer, I/O và thiết bị xử lý ngắt.
* Đặc điểm của VĐK 8051
  + Xung nhịp 12 MHz. Thời gian chu kỳ lệnh của bộ xử lý 1 μs.
  + ALU-8bit
  + Kiến trúc bộ nhớ Harvard
  + Internal-bus dữ liệu 8-bit và Internal-bus địa chỉ16-bit
  + Kiến trúc tập lệnh CISC
  + Không xử lý dấu phẩy động,
  + Không có bộ đệm Cache,
  + Không có đơn vị quản lý bộ nhớ MMU,
  + Không có đơn vị Thao tác nguyên tử (atomic operations unit)
  + Không có kỹ thuật đường ống CPU Pipeline
  + Không xử lý các lệnh song song.
  + RAM kích thước 128bytes
  + 32 byte RAM được sử dụng làm bốn bộ thanh ghi (register-set / bank). Mỗi bộ thanh ghi (bank) có 8 thanh ghi.
  + Bộ nhớ ngăn xếp/Bộ nhớ dữ liệu ngoài có thể được thêm tối đa 64 kB.
* Tín hiệu điều khiển EA
  + Là tín hiệu mà khi nó được kích hoạt ─ bộ xử lý luôn truy cập các địa chỉ ngoài bộ nhớ thay vì bộ nhớ trong hoặc địa chỉ thanh ghi

**Câu 9:**

* Trạng thái máy: Được định nghĩa là thời gian một chu kỳ xung đồng hồ hệ thống CLK
* Chu kỳ máy: Là tập hợp của một số trạng thái máy để CPU hay một vi mạch khi nắm quyền kiểm soát BUS hệ thống, thực hiện xong một thao tác trên BUS hệ thống.
* Chu kỳ lệnh: Là tập hợp các chu kì máy cần thiết để hoàn thành một lệnh của máy.

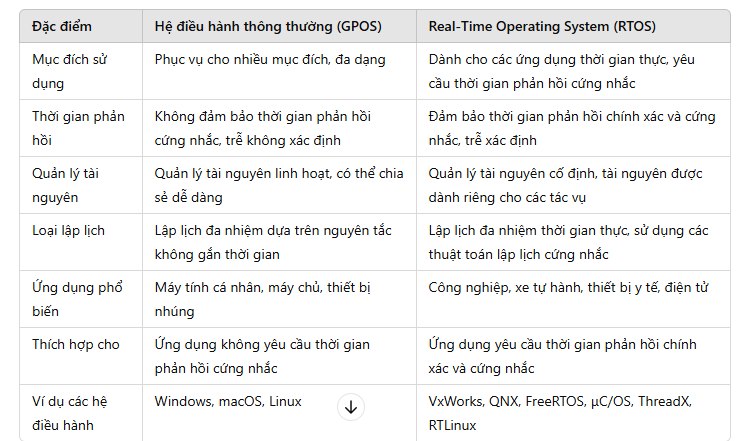
Chương 3:

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.
* 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 device 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 khai 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, ổ đĩa SSD, ổ đĩa 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 các ký tự hoặc byte riêng lẻ.
    - **Ví dụ**: Driver cho thiết bị giao tiếp qua cổng serial (UART), thiết bị âm thanh, thiết bị nhập liệu như bàn phím hoặc chuột.
    - **Đặc điểm**: Thường 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 như tty (teletypewriter), console.
    - **Đặc điểm**: Xử lý các tác vụ như nhập xuất dữ liệu, quản lý phiên 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 qua 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.

1. Real Time Operating System (RTOS)

* Định nghĩa:
  + Hệ điều hành thời gian thực là hệ điều hành **đa tác vụ** cho các ứng dụng với các ràng buộc (constraints) **thời gian thực**, được dùng trong lĩnh vực thiết kế các hệ thống nhúng.
  + Ràng buộc về thời gian thực: yêu cầu một độ trễ nhất định cho một sự kiện và đáp ứng của hệ thống.
* Đặc điểm
  + Một thuộc tính quan trọng của RTOS là khả năng tách biệt với ứng dụng. Nếu một chương trình bị “chết” hay hoạt động không hợp lệ, hệ điều hành thời gian thực có thể:
    - Cô lập chương trình này
    - Kích hoạt cơ chế phục hồi
    - Bảo vệ các chương trình khác
* Cơ chế hoạt động
  + RTOS có nhiệm vụ giải quyết, điều phối các tác vụ, tạo bộ lịch và phân mức ưu tiên. Tất cả các việc này được thực hiện nhờ cơ chế:
    - Hướng sự kiện (Event-driven): giải quyết, điều phối thông qua mức ưu tiên của tác vụ
    - Chia sẻ thời gian (Time-sharing): chuyển đổi các tác vụ dựa trên cơ chế ngắt
* Các chức năng cơ bản:
  + 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
  + Bộ lập lịch
  + Đồng bộ và thông điệp.
* Phân loại
  + Hard RTOS
    - có khả năng đáp ứng các yêu cầu về thời gian thực trong mọi trường hợp.
    - Các ứng dung sử dung hard RTOS thường là thiết bị yêu cầu xử lý ngay lập tức như xe tự lái, hệ thống tự động trên máy bay, tên lửa …
* Soft RTOS
  + đảm bảo đáp ứng các yêu cầu về thời gian thực nhưng có thể bị trễ trong một số tình huống nhất định.
* So sánh RTOS và OS



1. Middleware

* nằm ở nhiều nơi, tách biệt với app, OS, driver.
* Thông thường năm trên Device Driver và OS dưới Application
* Định nghĩa:
  + xem như phần mềm trung gian, đảm bảo linh hoạt, an ninh và giúp liên kết các phần mềm ở các lớp kề cận.
  + Chứa các hàm dịch vụ, được gọi sử dụng nhiều lần bởi các phần mềm khác.

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

5. Các bước biên dịch phần mềm trong IDE

1. Tạo project
2. Cấu hình phần mềm biên dịch (chọn loại chip, 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

Counter và timer trong 8051

* Họ 8051 phiên bản ban đầu có hai bộ định thời T0 và T1
* Họ 8052 (phần mở rộng của 8051) có ba bộ định thời T0, T1 và T2

Timer/Counter hoạt động như một bộ đếm xung nhịp (xung clock). Nó chứa các thanh ghi chứa giá trị đếm và thanh ghi điều khiển hoạt động đếm này

Khi giá trị đếm vượt quá giá trị tối đa mà thanh ghi chứa giá trị của Timer/Counter thì xuất hiện hiện tượng gọi là “tràn (over flow)” và bắt đầu lại từ đầu, đồng thời tạo ra một cờ (flag) để báo hiệu sự kiện này.

Các mode:

- Mode 0: hoạt động ở chế độ 13 bit

- Mode 1: Hoạt động ở chế độ 16 bit

- Mode 2: Hoạt động ở chế độ 8-bit tự động nạp lại

- Mode 3: Hoạt động như hai Timer 8-bit độc lập

* Thanh ghi đặc biệt TMOD: cấu hình
* Thanh ghi đặc biệt TCON trên Bốn bit: điều khiển