

14. RUNNABLES

Khái niệm về Runnable

- **Runnable:** Là các đoạn mã nhỏ nhất được định nghĩa trên một thành phần phần mềm (SWC). Có thể hiểu đơn giản, runnable là các hàm riêng lẻ trên một thành phần phần mềm.
- **Chức năng:** Mỗi hàm trong tệp C phải được định nghĩa là runnable trong cấu hình Autosar và phải chỉ định các giao diện truy cập trong đó.

Quy tắc đặc biệt với Runnables

- **Không có runnable:** Thành phần phần mềm kết hợp (Composition software component) hoặc thành phần phần mềm tham số (Parameter software component) không có runnable vì chúng không có chức năng C file đi kèm.

Ví dụ minh họa

Giả sử có một tệp C trong Autosar định nghĩa 4 hàm: "Sum", "Difference", "Multiplication", và "Division". Các hàm này được coi là runnables và phải được định nghĩa trong cấu hình runnable.

- **Thành phần phần mềm (SWC_1):** Chứa các hàm "Sum", "Difference", "Multiplication", và "Division".
- **Runnable Configuration:** Định nghĩa các hàm này dưới dạng runnable trong cấu hình Autosar.

Cấu hình Runnable

- **Tag Runnable Entity:** Định nghĩa một runnable.
- **Data Read Access:** Định nghĩa các cổng nhận dữ liệu mà hàm sẽ đọc.
- **Data Write Access:** Định nghĩa các cổng cung cấp dữ liệu mà hàm sẽ ghi.

Ví dụ cấu hình:

```
<RunnableEntity>
  <ShortName>Sum</ShortName>
  <Symbol>Sum</Symbol>
  <DataReadAccess>
    <Variable>X</Variable>
    <Variable>Y</Variable>
  </DataReadAccess>
  <DataWriteAccess>
    <Variable>SumResult</Variable>
  </DataWriteAccess>
</RunnableEntity>
```

- **Symbol Tag:** Định nghĩa tên chính xác của hàm trong tệp C. Hệ điều hành sử dụng thông tin này để gọi hàm tương ứng.

Giao tiếp rõ ràng và ẩn

- **Giao tiếp rõ ràng (Explicit Communication):** Dữ liệu được truyền qua cổng như nó là, RTE không quan tâm đến tính nhất quán của thông điệp.
- **Giao tiếp ẩn (Implicit Communication):** RTE đảm bảo tính nhất quán của thông điệp giữa người gửi và nhiều người nhận bằng cách tạo bộ đệm thông điệp riêng biệt cho mỗi người nhận.

Các loại cấu hình giao tiếp

1. **Data Read/Write Access:** Được sử dụng cho giao tiếp ẩn.
2. **Data Received by Argument/Value:** Được sử dụng cho giao tiếp rõ ràng.
3. **Data Send Configuration:** Được sử dụng cho giao tiếp hàng đợi, RTE tạo hàng đợi thông điệp để lưu trữ dữ liệu từ người gửi và truyền cho người nhận khi được yêu cầu.

Cấu hình cho các giao diện khác

1. **Parameter Access:** Chỉ có thể đọc tham số hiệu chỉnh từ hàm.
2. **Mode Switch Ports:**
 - **Mode Access Point:** Truy cập chế độ.
 - **Mode Switch Point:** Thiết lập chế độ.
3. **Local Interface Access:** Được định nghĩa trong phạm vi của thành phần.
4. **Client Server Interface:**
 - **Synchronous Server Call Point:** Gọi hàm đồng bộ.
 - **Asynchronous Server Call Result Point:** Gọi hàm không đồng bộ và đọc kết quả sau.
5. **Triggers:**
 - **Internal Trigger:** Kích hoạt nội bộ.
 - **External Trigger:** Kích hoạt từ bên ngoài.

Các thuộc tính chung

- **Can be in concurrently:** Xác định nếu runnable có thể được gọi đệ quy hoặc song song trên các tác vụ khác nhau.
- **Symbol Tag:** Phải khớp chính xác với tên hàm trong tệp C để hệ điều hành biết hàm nào cần lập lịch cho runnable.

Kết luận

Cấu hình runnable phải chứa tất cả thông tin về các giao diện hoặc cổng mà nó truy cập cùng với bất kỳ cấu hình phụ nào đã đề cập. Runnables là thành phần quan trọng trong việc định nghĩa các chức năng nhỏ nhất và cách chúng tương tác với hệ thống trong Autosar.

15. EVENTS

Khái niệm về Events

Events là các cấu hình bổ sung cho Runnable, chỉ định cho hệ điều hành cách gọi hoặc lập lịch cho một Runnable. Các Runnable được ánh xạ tới các sự kiện (events) của RTE và hệ điều hành cùng với lớp RTE đảm bảo rằng hàm runnable được gọi một cách mong đợi. Khi một sự kiện xảy ra, hệ điều hành sẽ bắt đầu Runnable tương ứng.

Phân loại Events

Mặc dù Autosar không phân loại các sự kiện theo cách này, nhưng để dễ hiểu, chúng ta sẽ phân loại chúng dựa trên cách sử dụng.

1. Sự kiện chung (Generic Events):

- **Init Event:** Dùng để khởi tạo phần mềm, chỉ gọi một lần khi khởi động.
- **Timing Event:** Dùng khi cần gọi các runnable theo chu kỳ thời gian định kỳ.
- **External Trigger Occurred Event:** Kích hoạt khi một sự kiện bên ngoài xảy ra.
- **Internal Trigger Occurred Event:** Kích hoạt khi một sự kiện bên trong xảy ra.
- **Background Event:** Chạy các runnable ở chế độ nền khi CPU rảnh.

2. Sự kiện liên quan đến Client Server:

- **Operation Invoked Event:** Dùng phía server của giao tiếp khách hàng-máy chủ, kích hoạt khi một client yêu cầu thực hiện dịch vụ.
- **Asynchronous Server Result Event:** Kích hoạt khi một cuộc gọi server không đồng bộ hoàn thành.

3. Sự kiện liên quan đến dữ liệu:

- **Data Write Complete Event:** Kích hoạt khi dữ liệu đã được gửi trên cổng cung cấp.
- **Data Receive Event:** Kích hoạt khi dữ liệu được nhận trên cổng nhận.
- **Data Receive Error Event:** Kích hoạt khi có lỗi trong quá trình nhận dữ liệu.

4. Sự kiện liên quan đến chế độ (Mode Related Events):

- **Mode Switch Event:** Kích hoạt khi có sự thay đổi trạng thái trong một chế độ cụ thể.
- **Mode Manager Error Event:** Kích hoạt khi xảy ra lỗi trong quản lý chế độ.
- **Mode Switch Acknowledgement Event:** Kích hoạt khi các chế độ chuyển đổi được xác nhận.

Ví dụ minh họa

Giả sử có một thành phần phần mềm với 4 runnables: "Sum", "Difference", "Multiplication", và "Division". Mỗi runnable được ánh xạ tới một sự kiện để kích hoạt.

1. **Sum Runnable:** Ánh xạ tới sự kiện timing để gọi hàm định kỳ.
2. **Difference Runnable:** Ánh xạ tới sự kiện operation invoked để gọi hàm khi client yêu cầu dịch vụ.

3. **Multiplication Runnable:** Ảnh xạ tới sự kiện data received để gọi hàm khi có dữ liệu mới được nhận.
4. **Division Runnable:** Ảnh xạ tới sự kiện mode switch để gọi hàm khi một chế độ cụ thể được kích hoạt.

Cấu hình Events

Ví dụ cấu hình:

```
<Events>
  <Event>
    <Type>TimingEvent</Type>
    <Runnable>Sum</Runnable>
    <Period>0.1</Period>
  </Event>
  <Event>
    <Type>OperationInvokedEvent</Type>
    <Runnable>Difference</Runnable>
    <ProviderPort>ClientServerProviderPort</ProviderPort>
  </Event>
  <Event>
    <Type>DataReceivedEvent</Type>
    <Runnable>Multiplication</Runnable>
    <ReceiverPort>DataReceiverPort</ReceiverPort>
  </Event>
  <Event>
    <Type>ModeSwitchEvent</Type>
    <Runnable>Division</Runnable>
    <Mode>HeadLightMode</Mode>
  </Event>
</Events>
```

1. **Timing Event:**
 - Gọi runnable "Sum" mỗi 100ms.
 - **Period:** Thời gian gọi runnable được đề cập dưới dạng giây (0.1 giây tương đương 100ms).
2. **Init Event:**
 - Gọi runnable chỉ một lần khi khởi tạo.
 - Không có thuộc tính period.
3. **Data Received Event:**
 - Gọi runnable khi nhận được dữ liệu trên cổng nhận.
 - **ReceiverPort:** Cổng nhận dữ liệu.
4. **Operation Invoked Event:**
 - Gọi runnable khi client server yêu cầu thực hiện dịch vụ.
 - **ProviderPort:** Cổng cung cấp của client server interface.
5. **Mode Switch Event:**
 - Gọi runnable khi có sự thay đổi chế độ.
 - **Mode:** Chế độ liên quan (ví dụ: HeadLightMode).

Kết luận

Việc ánh xạ runnables tới các sự kiện giúp hệ điều hành và lớp RTE quản lý và gọi các hàm runnable một cách hiệu quả và chính xác. Các sự kiện khác nhau cung cấp các cách khác nhau để kích hoạt runnables dựa trên yêu cầu cụ thể của phần mềm.