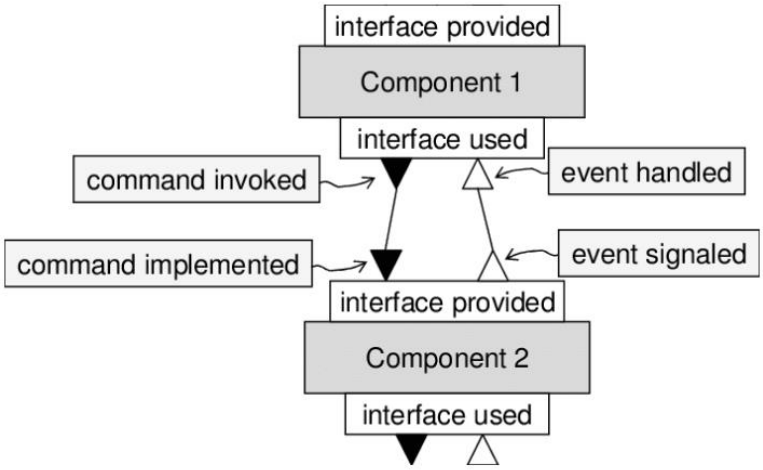
Exploring and understanding TinyOS computational concepts: - Events, Commands and Task.

(nesC model, nesC Components)

Aim:

To understand TinyOS computational concepts through nesC programming language

Theory:



NesC (Network embedded systems C)

A NesC is a programming language that was developed specifically for wireless sensor networks (WSNs). It was created at the University of California, Berkeley, and is based on the C programming language.

NesC is designed to be a component-based language, meaning that it is structured around modular components that can be combined to form larger programs. This makes it well-suited for developing applications for WSNs, which often involve a large number of small, specialized devices working together.

One of the key features of NesC is its support for a programming model known as "event-driven programming". In this model, programs are structured around events and the handlers that respond to them. This approach is particularly useful in WSNs, where devices often need to respond to events such as changes in the environment or changes in the network topology.

NesC also includes features for low-level programming, such as direct memory access and hardware control, which are important in the resource-constrained environment of WSNs.

Overall, NesC is a powerful and flexible programming language that is specifically designed for the unique challenges of developing applications for wireless sensor networks.

NesC Model

The NesC model refers to the programming model used in the NesC programming language. The NesC model is based on a component-based architecture, in which programs are constructed from reusable software components.

In the NesC model, a component is a self-contained module that performs a specific function or provides a specific service. Components can be combined to create larger programs, and the connections between components are defined by interfaces. Interfaces define the methods and signals that are used to interact with a component, and provide a way for components to communicate with each other.

The NesC model also includes support for event-driven programming, which is a programming paradigm in which programs are structured around events and the handlers that respond to them. In the NesC model, events are defined as signals, and the handlers that respond to them are defined as tasks.

Overall, the NesC model provides a powerful and flexible way to develop complex programs for wireless sensor networks by using modular, reusable components and an event-driven programming approach.

NesC Components

In the NesC programming language, a component is a self-contained module that performs a specific function or provides a specific service. NesC components are designed to be highly modular and reusable, which makes it easy to create complex programs by combining and connecting different components together.

Some common types of components in NesC include:

1) Modules: These are the basic building blocks of NesC programs. A module is a self-contained unit of code that defines a single functionality or a single interface. Modules can include functions, variables, and interfaces that other modules can use.

2) Configurations: A configuration is a group of modules and their interconnections that make up a complete program. Configurations define how modules are connected to each other and how they interact to achieve a specific task.

3) Interfaces: Interfaces define the methods and signals that are used to interact with a component. They provide a way for components to communicate with each other, and allow for components to be developed independently and easily swapped out or modified as needed.

4) Components Libraries: Components libraries provide pre-built and reusable components that can be easily included in a program. They can be used to simplify programming tasks, speed up development, and ensure the quality of the code.

5) Libraries: NesC also supports the use of standard C libraries, which can be used to provide additional functionality to NesC programs. Standard C libraries can be used to perform tasks such as memory allocation, string manipulation, and math operations.

Overall, NesC components provide a powerful way to create modular, reusable code that can be easily combined and connected to build complex programs for wireless sensor networks.