**1. Fork/ Join là gì?** là một framework được thêm vào java 7. cung cấp các công cụ giúp tăng tốc xử lý song song bằng cách cố gắng sử dụng tất cả các lõi bộ xử lý có sẵn, được thực hiện thông qua cách tiếp cận phân chia (fork) và gộp (join) task.

**2. Mục đích:** để sử dụng tất cả các khả năng xử lý để nâng cao hiệu suất cho các ứng dụng.

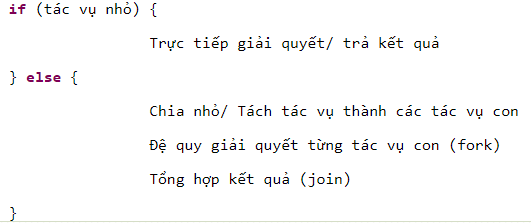
**3. Fork/ Join làm việc như thế nào?**

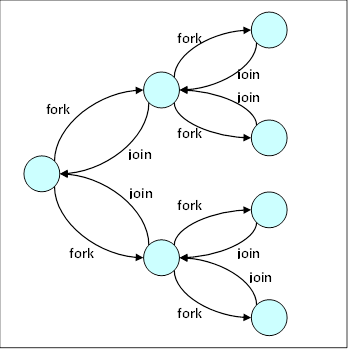
Gồm 2 bước:

+ Đầu tiên framework Fork/ Join thực hiện là chia nhỏ task (fork/ split), đệ quy chia nhỏ nhiệm vụ thành các nhiệm vụ phụ nhỏ hơn cho đến khi chúng đơn giản đủ để được thực hiện xử lý không đồng bộ.

+ Phần gộp kết quả (join) bắt đầu, trong đó các kết quả của tất cả các nhiệm vụ phụ được đệ quy một cách đệ quy vào một kết quả, hoặc trong trường hợp một nhiệm vụ trả về void, chương trình chỉ cần đợi cho đến khi mỗi nhiệm vụ phụ được thực hiện.

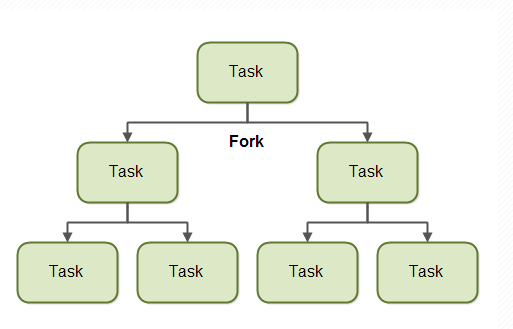
+ Cụ thể:





**4. Nguyên tắc hoạt động của Fork/Join**

**4.1 Nguyên tắc hoạt động của Fork.**



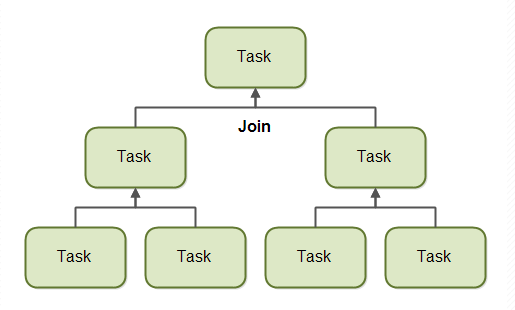
Theo sơ đồ, một nhiệm vụ sử dụng nguyên tắc Fork/Join có thể chia tách chính nó vào các nhiệm vụ con nhỏ hơn để có thể thực hiện đồng thời.

Bằng cách chia nhỏ thành các nhiệm vụ con như vậy, mỗi nhiệm vụ con có thể được thực hiện song song bởi các CPU khác nhau, hoặc các luồng khác nhau trên cùng một CPU.

- Điều kiện để phân tách thành các nhiệm vụ phụ:

Nếu công việc mà nhiệm vụ được đưa ra là đủ lớn để điều này có ý nghĩa. Có một chi phí để chia tách một nhiệm vụ thành các nhiệm vụ phụ, vì vậy với số lượng nhỏ công việc trên không thể lớn hơn tốc độ đạt được bằng cách thực hiện các công việc phụ đồng thời.

**4.2 Nguyên tắc hoạt động của Join**



Theo sơ đồ, các nhiệm vụ con được tách từ 1 nhiệm vụ cha thì nhiệm vụ cha đó sẽ đợi cho đến khi các nhiệm vụ con hoàn thành.

Khi nhiệm vụ con đã hoàn thành, nhiệm vụ cha có thể kết hợp tất cả các kết quả con vào một kết quả cuối cùng. Nếu các nhiệm vụ con không trả về kết quả thì nhiệm vụ cha chỉ cần đợi các nhiệm vụ con hoàn thành và sự kết hợp sẽ không xảy ra.

**5. Hiệu năng**

Để cung cấp thc hiện xử lý song song hiệu quả, các Fork/Join sử dụng hồ chứa(pool) các Thread được gọi là ForkJoinPool.

ForkJoinPool tương tự như Java ExecutorService nhưng với một sự khác biệt. ForkJoinPool phân chia các tác vụ cho các luồng thực thi trong Thread Pool. Framework Fork/ Join sử dụng thuật toán work-stealing. Các luồng sẽ thực thi công việc của mình trên một bộ xử lý riêng biệt (thread/ processor), khi làm hết việc của mình, nó lấy bớt (steal) các tác vụ từ các luồng khác đang bận rộn.