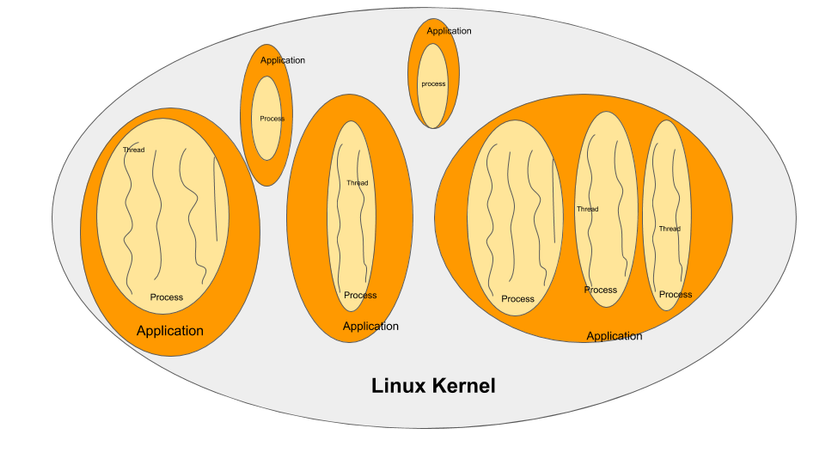
**Process trong Android**

Android có 4 components chính là ***Activity***, ***Service***, ***BroadcastReceiver***, ***ContentProvider***, khi một component của ứng dụng được khởi chạy mà trước đó ứng dụng không có bất kì component nào đang chạy, Hệ thống android khởi động một tiến trình (process) Linux cho ứng dụng với một luồng duy nhất.

Mặc định tất cả các component của cùng một ứng dụng sẽ được chạy trên một process và thread duy nhất (nó còn được gọi là Main Thread).

Các bạn có thể hiểu đơn giản rằng, Trong điện thoại có rất nhiều ứng dụng, mỗi ứng dụng lại có rất nhiều các thành phần, mỗi thành phần lại chạy trên các luồng khác nhau, và các luồng này thì được quản lý bằng các process khác nhau. Các bạn có thể hình dung qua hình sau:



* Mỗi Process cung cấp tài nguyên cần thiết để thực thi chương trình. Mỗi Process có một không gian địa chỉ ảo, có các mã thực thi, có các lệnh xử lý các đối tượng hệ thống, một ngữ cảnh bảo mật, kích thước làm việc tối thiểu và tối đa, ... và phải có ít nhất một Thread thực thi.
* Một Thread là một thực thể trong một Process, là đối tượng được lên kế hoạc để thực thi. Trong Process thường sẽ có nhiều Thread và tất cả các Thread này sẽ chia sẽ không gian địa chỉ ảo và tài nguyên hệ thống trong Process. Ngoài ra mỗi Thread lại có công việc riêng của nó đó là: duy trì xử lý ngoại lệ, ưu tiên lập lịch trình, lưu dữ cục bộ luồng,... Môi trường Thread có thể bao gồm: phần đăng kí Thread với máy chủ, nhân stack, ... quan trọng là có một môi trường (Thread cũng có ngữ cảnh riêng của nó).
* Tóm lại thì:

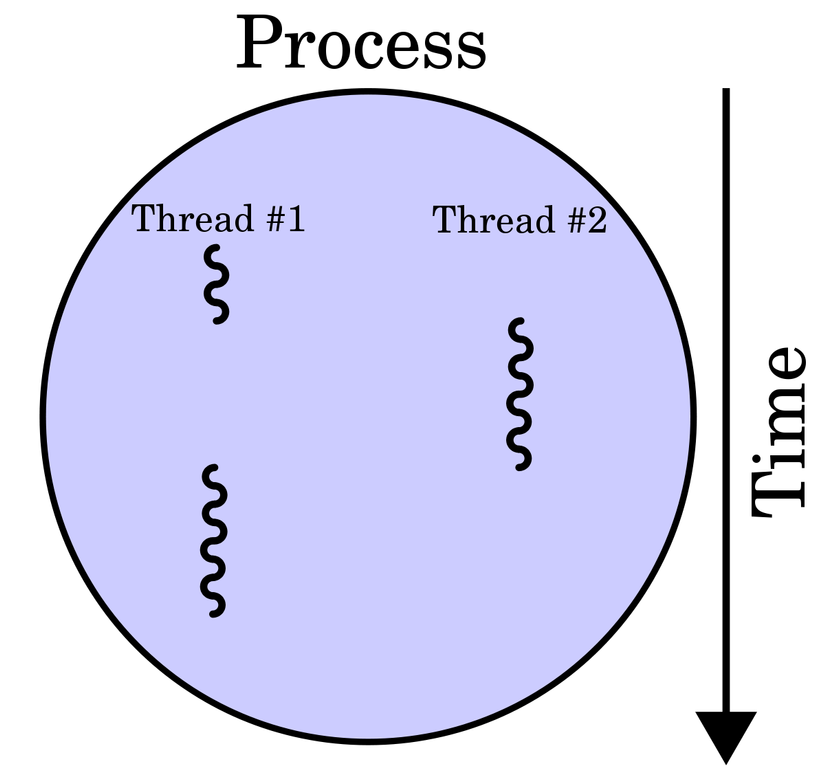
***Process = Program + State of all Threads executing in Program***

Có nghĩa là ***Process bao gồm chương trình cùng với tất cả trạng thái thực thi của các Thread trong chương trình đó***.

**Thread và MultiThread trong Android**

Vậy Thread là gì? Thread được định nghĩa là một luồng dùng để thực thi một chương trình. Java Virtual Machine cho phép một chương trình có thể có nhiều Thread thực thi đồng thời. Mỗi Thread đều có độ ưu tiên của nó.

Trong Android chúng ta có rất nhiều process,với mỗi một process chúng ta có thể tạo ra nhất nhiều thread trong đó, như hình sau:



Khi một ứng dụng được khởi chạy, một process được sinh ra (nếu chúng chưa được khởi chạy trước đó bởi bất kì component nào của ứng dụng), hệ thống sẽ tự động khởi tạo một thread cho ứng dụng trong process đó, gọi là "Main" Thread. Thread này vô cùng quan trọng vì nó phụ trách việc nhận tương tác từ phía giao diện người dùng, bao gồm cả việc vẽ các sự kiện và update lại các giao diện người dùng. Như vậy Main Thread đôi khi còn được gọi là UI Thread (sẽ có trường hợp ngoại lệ)

Hệ thống không tạo ra các thread riêng biệt cho từng instance của các component.Tất cả các component đều được chạy trên một process và được khởi tạo trên UI thread và các method xử lý của hệ thống đều được xử lý trên thread đó. Do đó các phuơng thức callback của hệ thống như (onKeyDown(), onStart(), onStop()...) đều được gọi về trên UI Thread của Process.

Khi ứng dụng của bạn thực hiện các tác vụ kéo dài để thực hiện các lệnh từ người dùng, việc chỉ sử dụng một luồng duy nhất trong Process mà hệ thống tạo ra sẽ gây rất chậm cho ứng dụng nhất là những tác vụ tuơng tác với internet. Việc block UI trong một khoảng thời gian dài (hiện tại là 5s) sẽ gây ra hiện tượng ANR (application not responding). Người dùng có thể quyết định tắt ứng dụng của các bạn.

Vì vậy chúng ta có những quy tắc khi sử dụng UI Thread như sau

1. Không được block UI Thead
2. Không được truy cập vào Android UI Tookit từ thread không phải UI Thread. (Không được update UI ngoài Main Thread)

Để tạo mới Thread ta có hai cách. Cách thứ nhất là kế thừa (extends) từ class Thread:

private class MyThread extends Thread{

@Override

public void run() {

//TODO

}

}

...

new MyThread().start();

Cách thứ 2 là thực thi (implements) interface Runnable:

private class MyRunnable implements Runnable{

@Override

public void run() {

//TODO

}

}

...

new Thread(new MyRunnable()).start();