

Herramientas/Lenguajes para hacer un daemon

Christofer Fabián Chávez Carazas

Universidad Nacional de San Agustín de Arequipa

Escuela Profesional de Ciencia de la Computación

Computación Centrada en Redes

22 de diciembre de 2017

■ Java

Existen 2 tipos de thread en java, los daemon y los no-daemon, cuando la JVM se inicia, por lo general hay un solo thread no-daemon y el que manda llamar al método `main ()` de un programa, y la JVM se ejecuta hasta que una llamada a `System.exit ()` se haga o todos threads que no son daemon hayan muerto. Los Daemon en Java son un proveedor de servicios para otros threads u objetos que se ejecutan en el mismo proceso en el que se ejecuta el daemon. Los daemon se utilizan para tareas secundarias y solo son necesarios cuando se ejecutan threads normales.

Un hilo recién creado hereda el estado del daemon de su padre. Esa es la razón por la cual todos los hilos creados dentro del método principal no son demonios de forma predeterminada, porque el hilo principal no es demonio. Sin embargo, se puede crear un hilo daemon utilizando el método `setDaemon()` de la clase `Thread`. Un ejemplo para crear un daemon es el siguiente:

```
public class DaemonThreadExample1 extends Thread{

    public void run(){

        // Checking whether the thread is Daemon or not
        if(Thread.currentThread().isDaemon()){
            System.out.println("Daemon thread executing");
        }
        else{
            System.out.println("user(normal) thread executing");
        }
    }

    public static void main(String[] args){
        /* Creating two threads: by default they are
        * user threads (non-daemon threads)
        */
        DaemonThreadExample1 t1=new DaemonThreadExample1 ();
        DaemonThreadExample1 t2=new DaemonThreadExample1 ();

        //Making user thread t1 to Daemon
        t1.setDaemon(true);

        //starting both the threads
        t1.start();
        t2.start();
    }
}
```

■ Ruby

Ruby es un poderoso lenguaje de programación dinámico. Puedes usarlo fácilmente para construir software grande y complejo. Sin embargo, hay ocasiones en las que solo se desea usarlo para crear un script básico para automatizar un flujo de trabajo. A partir de la versión 1.9 crear un daemon se ha vuelto un proceso bastante simple, no necesitas ninguna librería o gema adicional sólo debes agregar un código similar al siguiente:

```
# daemonize
Process.daemon(true,true)

# write pid to a .pid file
pid_file = File.dirname(__FILE__) + "#{__FILE__}.pid"
File.open(pid_file, 'w') { |f| f.write Process.pid }
```

La primera línea convierte el script a un proceso daemon. El método `daemon` acepta dos argumentos, el primer argumento controla el directorio de trabajo actual. Si se coloca falso el directorio de trabajo actual pasa a ser el root. El segundo argumento determina la salida.