



**Universidad Nacional de  
Quilmes  
Tecnicatura en Programación  
Informática**

**LABORATORIO DE SISTEMAS  
OPERATIVOS Y REDES**

**USO DE UN SISTEMA DE CONTROL DE  
PROCESOS: SUPERVISOR**

Informe de proyecto

**ALUMNOS:**

- Carlos Cuoco
- David Matías Leonhardt

**FECHA: 05/12/2013**

# Indice

[Objetivos](#)

[Alcance](#)

[Desarrollo](#)

[Instalación](#)

[Instalación desde repositorios](#)

[Instalación online usando python](#)

[Instalación manual](#)

[Configuración](#)

[Configuración principal](#)

[Configuración de programas](#)

[Configuración de event listeners](#)

[Iniciar y detener supervisord y procesos](#)

[Administración web](#)

[Conclusiones](#)

[Referencias](#)

# Objetivos

Mediante el presente documento nos proponemos explicar el modo de instalación y configuración de un sistema de control de procesos, en particular **supervisord**.

## Alcance

Explicaremos la forma de instalación de supervisord en un sistema operativo Ubuntu 12.04. Mostraremos como configurar supervisord. También mostraremos cómo iniciar y detener tanto al mismo supervisor como a los procesos que él administre desde consola y de forma web con la interfaz que él mismo provee.

## Desarrollo

### Instalación

#### Instalación desde repositorios

La forma más sencilla de instalación es desde los repositorios de Ubuntu con el siguiente comando:

```
sudo apt-get install supervisor
```

A pesar de ser la manera más sencilla, es desaconsejable ya que se instalará una versión desactualizada (del año 2010).

#### Instalación online usando python

El software necesario para la utilización de supervisord en Ubuntu es Python 2.4 en adelante, no funcionando con Python 3.0 en adelante.

Una vez que esta instalado Python, instalaremos *setuptools* (un compendio de herramientas para facilitar la instalación de módulos en Python) .

```
apt-get install python-setuptools
```

Una vez que instalamos las `setuptools`, utilizaremos una función de la misma:

```
easy_install supervisor
```

Tener en cuenta que para estas invocaciones puede ser necesario tener permisos de administrador

## Instalación manual

En caso de tener que instalar sin conexión a Internet también es necesario

- `meld3` ([http://www.plope.com/software/meld3/.](http://www.plope.com/software/meld3/))
- `elementtree` ([http://effbot.org/downloads#elementtree.](http://effbot.org/downloads#elementtree))

Invocando la línea:

```
python setup.py install
```

## Configuración

### Configuración principal

Para conocer la configuración por defecto podemos ejecutar el siguiente comando desde la consola:

```
echo_supervisord_conf
```

Conociendo dicha configuración podemos ejecutar el siguiente comando para generar el verdadero archivo que va a contener nuestra configuración:

```
echo_supervisord_conf > /etc/supervisord.conf
```

Por ahora descartamos la configuración web y nos centramos en la configuración mediante los archivos de texto.

De esta forma, tenemos dos maneras de configurar `supervisord` para que ejecute nuestros procesos. Una forma es configurarlos todos en el archivo recién creado (`/etc/supervisord.conf`), el cual crecerá a medida que incluyamos procesos para ser supervisados, por ese motivo no es recomendable esta opción.

La otra opción, y recomendable, es dividir la configuración en pequeños archivos cada uno con la configuración particular de cada proceso. Para eso, primero tenemos que modificar el archivo `supervisord.conf` para que acepte esta metodología de trabajo.

Si editamos `/etc/supervisord.conf`, al final del mismo encontramos lo siguiente:

```
;[include]  
;files = relative/directory/*.ini
```

Lo cambiamos por lo siguiente:

```
[include]  
files = /etc/supervisor/*.conf
```

Debido al cambio que acabamos de hacer, es necesario que creamos la carpeta supervisor dentro de /etc:

```
cd /etc  
mkdir supervisor
```

Ahora es posible tener separada nuestras configuraciones para cada proceso. Dichas configuraciones van a estar en el directorio /etc/supervisor con la extensión .conf. Es recomendable que el nombre del archivo sea el nombre del proceso a supervisar para una fácil y rápida identificación.

La configuración principal de supervisor está bajo el tag [supervisord].

Las principales configuraciones son las siguientes:

- logfile=/tmp/supervisord.log  
Nombre y ubicación del archivo de log
- logfile\_maxbytes=50MB  
Tamaño máximo de los archivos de log
- logfile\_backups=10  
Cantidad máxima de archivos de log
- loglevel=info  
Nivel del log (puede ser error, debug, warning, info, trace)
- pidfile=/tmp/supervisord.pid  
Lugar y nombre del archivo que contendrá el número pid del sistema operativo
- directory=/tmp  
Directorio donde se guardarán archivos generados por supervisor, por ej.: archivo de error.

## Configuración de programas

Dentro del archivo \*.conf que creamos dentro de /etc/supervisor tenemos que completarlo con alguna de las siguientes opciones (todos los tiempos son en segundos):

<b>[program:nombreDelPrograma]</b>	
command=/bin/cat	comando a ejecutar
process_name=%(program_name)s	nombre del proceso (default %(program_name)s)
numprocs=1	cantidad de procesos iguales a ejecutar (def 1)
directory=/tmp	directorio donde ejecutar el comando (default sin directorio)
umask=022	permisos para el proceso (default ninguno)

priority=999	prioridad (default 999)
autostart=true	; inicia al ejecutarse supervisord (default: true)
autorestart=unexpected	cuando reiniciar (default: unexpected, puede ser false o true)
startsecs=1	segundos necesarios para que el proceso esté activo (def. 1)
startretries=3	cantidad de reintentos para iniciar el proceso (default 3)
exitcodes=0,2	exit codes generados por el proceso (default 0,2)
stopsignal=QUIT	señal usada para terminar el proceso (default TERM)
stopwaitsecs=10	tiempo máximo para esperar que el procesos finalice (default 10)
stopasgroup=false	finaliza procesos hijos (default false)
killasgroup=false	mata procesos hijos (def false)
user=chrism	usuario UNIX que ejecutará el proceso (default root)
redirect_stderr=true	redirect proc stderr to stdout (default false)
stdout_logfile=/a/path	directorio donde guardar el log stdout, NONE for none; default AUTO
stdout_logfile_maxbytes=1MB	tamaño del log stdout (default 50MB)
stdout_logfile_backups=10	cantidad de archivos de stdout (default 10)
stdout_capture_maxbytes=1MB	tamaño en 'capturemode' (default 0)
stdout_events_enabled=false	emisión de eventos cuando se escribe en stdout (default false)
stderr_logfile=/a/path	directorio donde guardar el log stderr, NONE for none; default AUTO
stderr_logfile_maxbytes=1MB	tamaño del log stderr (default 50MB)
stderr_logfile_backups=10	cantidad de archivos de stderr (default 10)
stderr_capture_maxbytes=1MB	tamaño en 'capturemode' (default 0)
stderr_events_enabled=false	emisión de eventos cuando se escribe en stderr (default false)
environment=A="1",B="2"	variables de proceso (def no vars)
serverurl=AUTO	sobreescribir url (childutils)

Para mayor información, chequear este link:

<http://supervisord.org/configuration.html#programx-section>

Un ejemplo podría ser el siguiente:

```
[program:holidays]
user=user
autostart=true ; este parámetro podría obviarse ya que true es el valor por defecto
autorestart=true
command = /home/user/holidays/start
```

## Configuración de event listeners

La configuración para esta opción es casi idéntica a la de programas, cambiando algunos valores por defecto. Para mayor información revisar este link:

<http://supervisord.org/configuration.html#eventlistener-x-section-settings>

La diferencia que tiene con respecto a un programa, es que el comando configurado en la variable `command` solo se ejecutará cuando ocurra el evento seteado en la variable `event`.

Un ejemplo podría ser el siguiente:

```
[eventlistener:checkstate]  
command=python /opt/checkState.py  
events=PROCESS_STATE_STOPPING
```

## Iniciar y detener supervisor y procesos

La forma más sencilla de iniciar supervisor es ejecutar el siguiente comando:

```
sudo /usr/local/bin/supervisord
```

El único problema al usar este comando es que no tenemos forma de detenerlo ni reiniciarlo, con lo cuál es necesario utilizar un **kill** para matar el proceso y debemos hacer lo mismo por cada programa que supervisor haya ejecutado.

Afortunadamente tenemos personas que contribuyeron en este aspecto, en el siguiente link se pueden descargar scripts para controlar supervisor para distintos sistemas operativos:

<https://github.com/Supervisor/initscripts>

En particular usaremos el de ubuntu:

<https://github.com/Supervisor/initscripts/blob/master/ubuntu>

Teniendo el script, primero necesitamos darle permisos de ejecución:

```
sudo chmod +x /rutaAlArchivo/nombreDelArchivo
```

Lo ideal es que dicho archivo se llame `supervisord` y esté en `/etc/init.d`, de esta forma lo podemos controlar como a todos los demás servicios usando el comando **service**.

Algunos detalles para utilizar el script dado que colocamos los archivos en las ubicaciones por default:

- El script utiliza un conjunto de librerías denominadas `lsb`, la podemos instalar con **sudo apt-get install lsb**
- La variable `DAEMON` setearla en `/usr/local/bin/supervisord`
- La variable `DAEMON_OPTS` setearla en `"-c /etc/supervisord.conf $DAEMON_OPTS"`

Ahora si, podemos ejecutar el script:

- **`sudo /rutaAlArchivo/nombreDelArchivo start >>>>`** para iniciar supervisor
- **`sudo /rutaAlArchivo/nombreDelArchivo stop >>>>`** para detener supervisor
- **`sudo /rutaAlArchivo/nombreDelArchivo restart >>>>`** para reiniciar supervisor
- **`sudo /rutaAlArchivo/nombreDelArchivo reload >>>>`** para recargar la configuración de supervisor

Con lo anterior podemos controlar a supervisord en sí. Para controlar a cada programa existe el comando **`supervisorctl`**:

- **`sudo supervisorctl start nombrePrograma >>>>`** para iniciar un programa en particular
- **`sudo supervisorctl stop nombrePrograma >>>>`** para detener un programa en particular
- **`sudo supervisorctl restart nombrePrograma >>>>`** para reiniciar un programa en particular

## Administración web

Dentro del archivo `supervisord.conf`, hay que hacer dos cambios para poder habilitar la interfaz web. Por un lado, en la sección `supervisorctl` hay que descomentar la línea que hace referencia a la url del servidor:

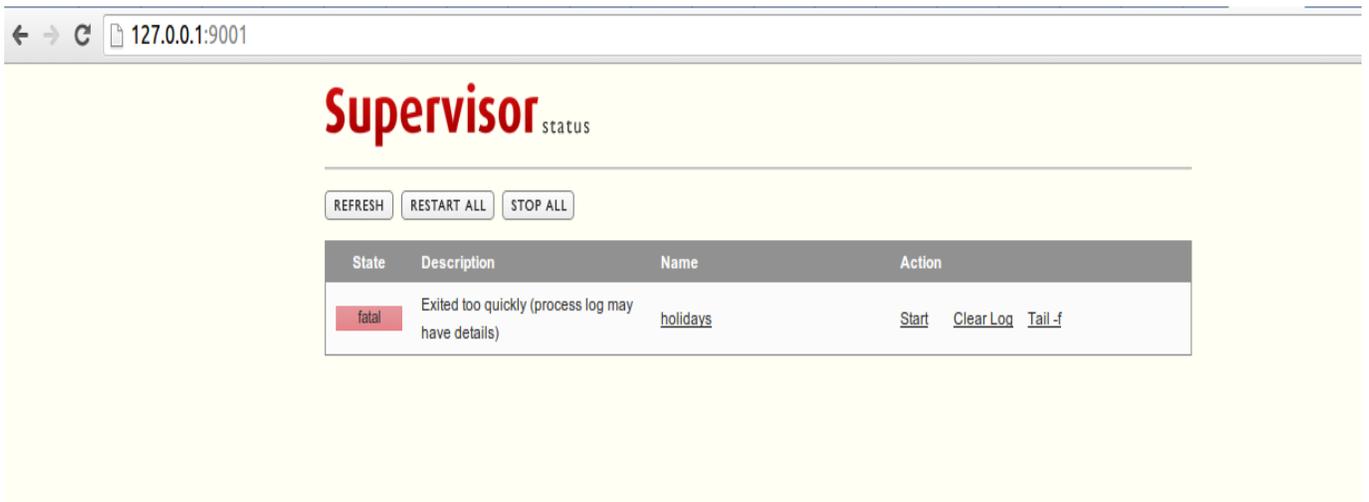
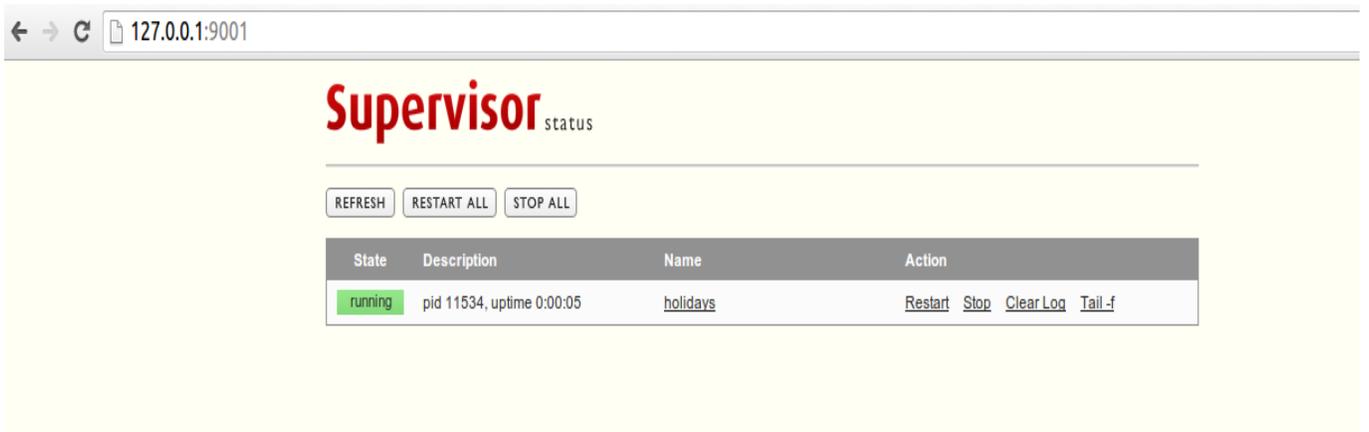
```
[supervisorctl]
;serverurl=unix:///tmp/supervisor.sock ; use a unix:// URL for a unix socket
serverurl=http://127.0.0.1:9001 ; use an http:// url to specify an inet socket
;username=chris ; should be same as http_username if set
;password=123 ; should be same as http_password if set
;prompt=mysupervisor ; cmd line prompt (default "supervisor")
;history_file=~/.sc_history ; use readline history if available
```

Adicionalmente hay que descomentar el siguiente bloque:

```
[inet_http_server] ; inet (TCP) server disabled by default
port=127.0.0.1:9001 ; (ip_address:port specifier, *:port for all iface)
username=user ; (default is no username (open server))
password=123 ; (default is no password (open server))
```

Aquí podemos configurar el puerto donde está la interfaz web además del usuario y contraseña.

Una vez configurado y reiniciado `supervisord`, podemos entrar desde nuestro navegador y veremos algo similar a las siguientes imágenes:



En la interfaz se puede reiniciar, detener o iniciar todos los procesos, como cada uno de forma individual. Además de cada proceso que se esté monitoreando se pueden ver los logs haciendo click sobre el link con el nombre del proceso.

## Conclusiones

Supervisord es una herramienta de gran utilidad para mantener siempre corriendo nuestras aplicaciones sin necesidad de tener que crear nuestro propio demonio para llevar a cabo dicha tarea. Esto nos quita un trabajo importante, haciendo que podamos centrarnos en nuestra aplicación y no invirtiendo tiempo en como haremos para que se reinicie si por algún motivo falla.

## Referencias

- Página oficial de supervisord: <http://supervisord.org/>