[unix\_http\_server]

file=/var/run/supervisor/supervisor.sock ; (the path to the socket file)

[supervisord]

logfile=/var/log/supervisor/supervisord.log ; (main log file;default $CWD/supervisord.log)

logfile\_maxbytes=50MB ; (max main logfile bytes b4 rotation;default 50MB)

logfile\_backups=10 ; (num of main logfile rotation backups;default 10)

loglevel=info ; (log level;default info; others: debug,warn,trace)

pidfile=/var/run/supervisord.pid ; (supervisord pidfile;default supervisord.pid)

nodaemon=false ; (start in foreground if true;default false)

minfds=1024 ; (min. avail startup file descriptors;default 1024)

minprocs=200 ; (min. avail process descriptors;default 200)

[rpcinterface:supervisor]

supervisor.rpcinterface\_factory = supervisor.rpcinterface:make\_main\_rpcinterface

[supervisorctl]

serverurl=unix:///var/run/supervisor/supervisor.sock ; use a unix:// URL for a unix socket

[include]

files = supervisord.d/test/\*.ini

原文：

; Sample supervisor config file.

[unix\_http\_server]

file=/var/run/supervisor/supervisor.sock ; (the path to the socket file)

;chmod=0700 ; sockef file mode (default 0700)

;chown=nobody:nogroup ; socket file uid:gid owner

;username=user ; (default is no username (open server))

;password=123 ; (default is no password (open server))

;[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))

[supervisord]

logfile=/var/log/supervisor/supervisord.log ; (main log file;default $CWD/supervisord.log)

logfile\_maxbytes=50MB ; (max main logfile bytes b4 rotation;default 50MB)

logfile\_backups=10 ; (num of main logfile rotation backups;default 10)

loglevel=info ; (log level;default info; others: debug,warn,trace)

pidfile=/var/run/supervisord.pid ; (supervisord pidfile;default supervisord.pid)

nodaemon=false ; (start in foreground if true;default false)

minfds=1024 ; (min. avail startup file descriptors;default 1024)

minprocs=200 ; (min. avail process descriptors;default 200)

;umask=022 ; (process file creation umask;default 022)

;user=chrism ; (default is current user, required if root)

;identifier=supervisor ; (supervisord identifier, default is 'supervisor')

;directory=/tmp ; (default is not to cd during start)

;nocleanup=true ; (don't clean up tempfiles at start;default false)

;childlogdir=/tmp ; ('AUTO' child log dir, default $TEMP)

;environment=KEY=value ; (key value pairs to add to environment)

;strip\_ansi=false ; (strip ansi escape codes in logs; def. false)

; the below section must remain in the config file for RPC

; (supervisorctl/web interface) to work, additional interfaces may be

; added by defining them in separate rpcinterface: sections

[rpcinterface:supervisor]

supervisor.rpcinterface\_factory = supervisor.rpcinterface:make\_main\_rpcinterface

[supervisorctl]

serverurl=unix:///var/run/supervisor/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

; The below sample program section shows all possible program subsection values,

; create one or more 'real' program: sections to be able to control them under

; supervisor.

;[program:theprogramname]

;command=/bin/cat ; the program (relative uses PATH, can take args)

;process\_name=%(program\_name)s ; process\_name expr (default %(program\_name)s)

;numprocs=1 ; number of processes copies to start (def 1)

;directory=/tmp ; directory to cwd to before exec (def no cwd)

;umask=022 ; umask for process (default None)

;priority=999 ; the relative start priority (default 999)

;autostart=true ; start at supervisord start (default: true)

;autorestart=true ; retstart at unexpected quit (default: true)

;startsecs=10 ; number of secs prog must stay running (def. 1)

;startretries=3 ; max # of serial start failures (default 3)

;exitcodes=0,2 ; 'expected' exit codes for process (default 0,2)

;stopsignal=QUIT ; signal used to kill process (default TERM)

;stopwaitsecs=10 ; max num secs to wait b4 SIGKILL (default 10)

;user=chrism ; setuid to this UNIX account to run the program

;redirect\_stderr=true ; redirect proc stderr to stdout (default false)

;stdout\_logfile=/a/path ; stdout log path, NONE for none; default AUTO

;stdout\_logfile\_maxbytes=1MB ; max # logfile bytes b4 rotation (default 50MB)

;stdout\_logfile\_backups=10 ; # of stdout logfile backups (default 10)

;stdout\_capture\_maxbytes=1MB ; number of bytes in 'capturemode' (default 0)

;stdout\_events\_enabled=false ; emit events on stdout writes (default false)

;stderr\_logfile=/a/path ; stderr log path, NONE for none; default AUTO

;stderr\_logfile\_maxbytes=1MB ; max # logfile bytes b4 rotation (default 50MB)

;stderr\_logfile\_backups=10 ; # of stderr logfile backups (default 10)

;stderr\_capture\_maxbytes=1MB ; number of bytes in 'capturemode' (default 0)

;stderr\_events\_enabled=false ; emit events on stderr writes (default false)

;environment=A=1,B=2 ; process environment additions (def no adds)

;serverurl=AUTO ; override serverurl computation (childutils)

; The below sample eventlistener section shows all possible

; eventlistener subsection values, create one or more 'real'

; eventlistener: sections to be able to handle event notifications

; sent by supervisor.

;[eventlistener:theeventlistenername]

;command=/bin/eventlistener ; the program (relative uses PATH, can take args)

;process\_name=%(program\_name)s ; process\_name expr (default %(program\_name)s)

;numprocs=1 ; number of processes copies to start (def 1)

;events=EVENT ; event notif. types to subscribe to (req'd)

;buffer\_size=10 ; event buffer queue size (default 10)

;directory=/tmp ; directory to cwd to before exec (def no cwd)

;umask=022 ; umask for process (default None)

;priority=-1 ; the relative start priority (default -1)

;autostart=true ; start at supervisord start (default: true)

;autorestart=unexpected ; restart at unexpected quit (default: unexpected)

;startsecs=10 ; number of secs prog must stay running (def. 1)

;startretries=3 ; max # of serial start failures (default 3)

;exitcodes=0,2 ; 'expected' exit codes for process (default 0,2)

;stopsignal=QUIT ; signal used to kill process (default TERM)

;stopwaitsecs=10 ; max num secs to wait b4 SIGKILL (default 10)

;user=chrism ; setuid to this UNIX account to run the program

;redirect\_stderr=true ; redirect proc stderr to stdout (default false)

;stdout\_logfile=/a/path ; stdout log path, NONE for none; default AUTO

;stdout\_logfile\_maxbytes=1MB ; max # logfile bytes b4 rotation (default 50MB)

;stdout\_logfile\_backups=10 ; # of stdout logfile backups (default 10)

;stdout\_events\_enabled=false ; emit events on stdout writes (default false)

;stderr\_logfile=/a/path ; stderr log path, NONE for none; default AUTO

;stderr\_logfile\_maxbytes=1MB ; max # logfile bytes b4 rotation (default 50MB)

;stderr\_logfile\_backups ; # of stderr logfile backups (default 10)

;stderr\_events\_enabled=false ; emit events on stderr writes (default false)

;environment=A=1,B=2 ; process environment additions

;serverurl=AUTO ; override serverurl computation (childutils)

; The below sample group section shows all possible group values,

; create one or more 'real' group: sections to create "heterogeneous"

; process groups.

;[group:thegroupname]

;programs=progname1,progname2 ; each refers to 'x' in [program:x] definitions

;priority=999 ; the relative start priority (default 999)

; The [include] section can just contain the "files" setting. This

; setting can list multiple files (separated by whitespace or

; newlines). It can also contain wildcards. The filenames are

; interpreted as relative to this file. Included files \*cannot\*

; include files themselves.

[include]

files = supervisord.d/test/\*.ini