

Ipmiwatchdog daemon

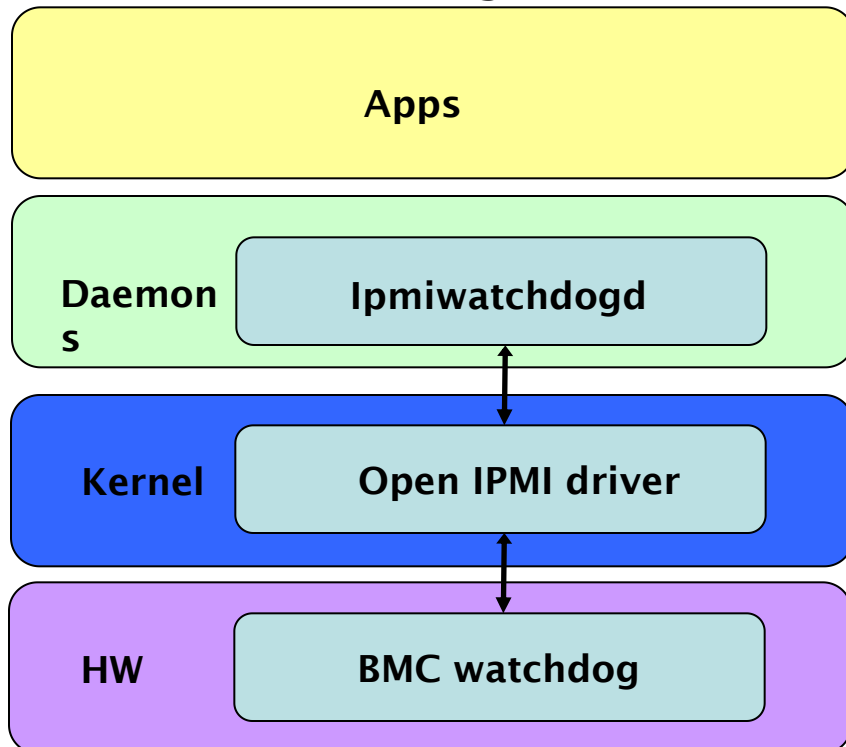
An ipmiwatchdog daemon implementation for ATCA blades

By Sam Lee

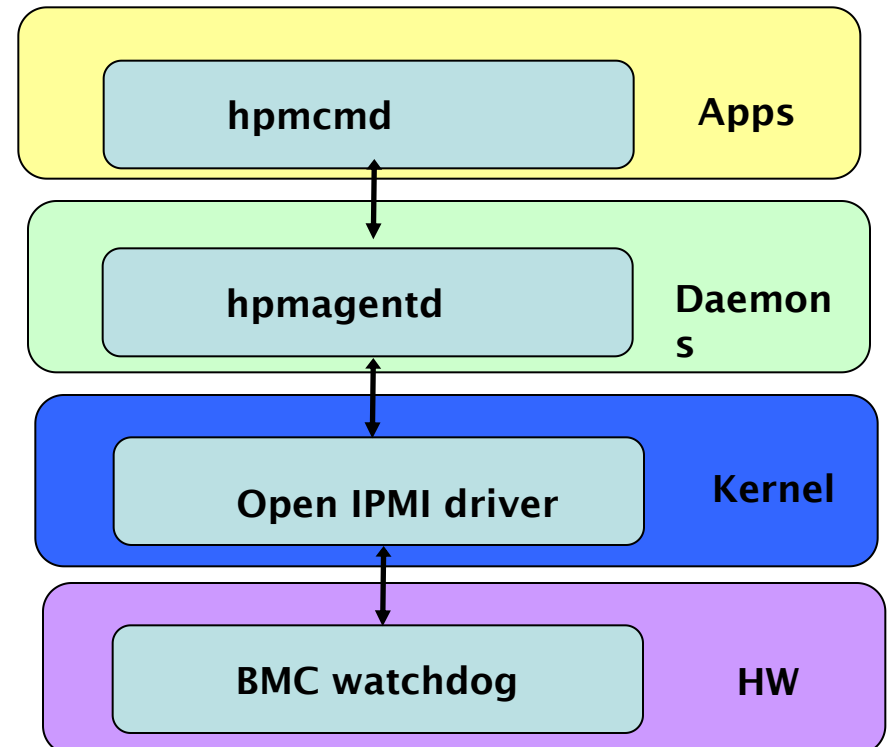
ipmiwatchdogd-advance

- 1, We used to control the BMC watchdog thru the hpmcmd, but sometimes, it's not stable enough.
- 2, We may control the BMC thru software watchdog daemon, which requires ipmi_watchdog.ko and WATCHDOG [=y] compiled in kernel", and the software watchdog daemon uses to control device file "/dev/watchdog". The file "/dev/watchdog" is shared by software watchdog, TSC watchdog, and other watchdogs. So if the file is used to control BMC watchdog by software watchdog daemon, that means all of the other watchdogs can not be any more.
- 3, So based on above reasons(1,2), I wrote this Ipmiwatchdogd. This daemon uses "/dev/ipmi0" which BMC dedicated. So all of the other watchdogd can be used at the same time for redundancy. And it as an daemon and warps raw IPMI commands to control BMC watchdog, that more stable than hpmcmd watchdog commands.

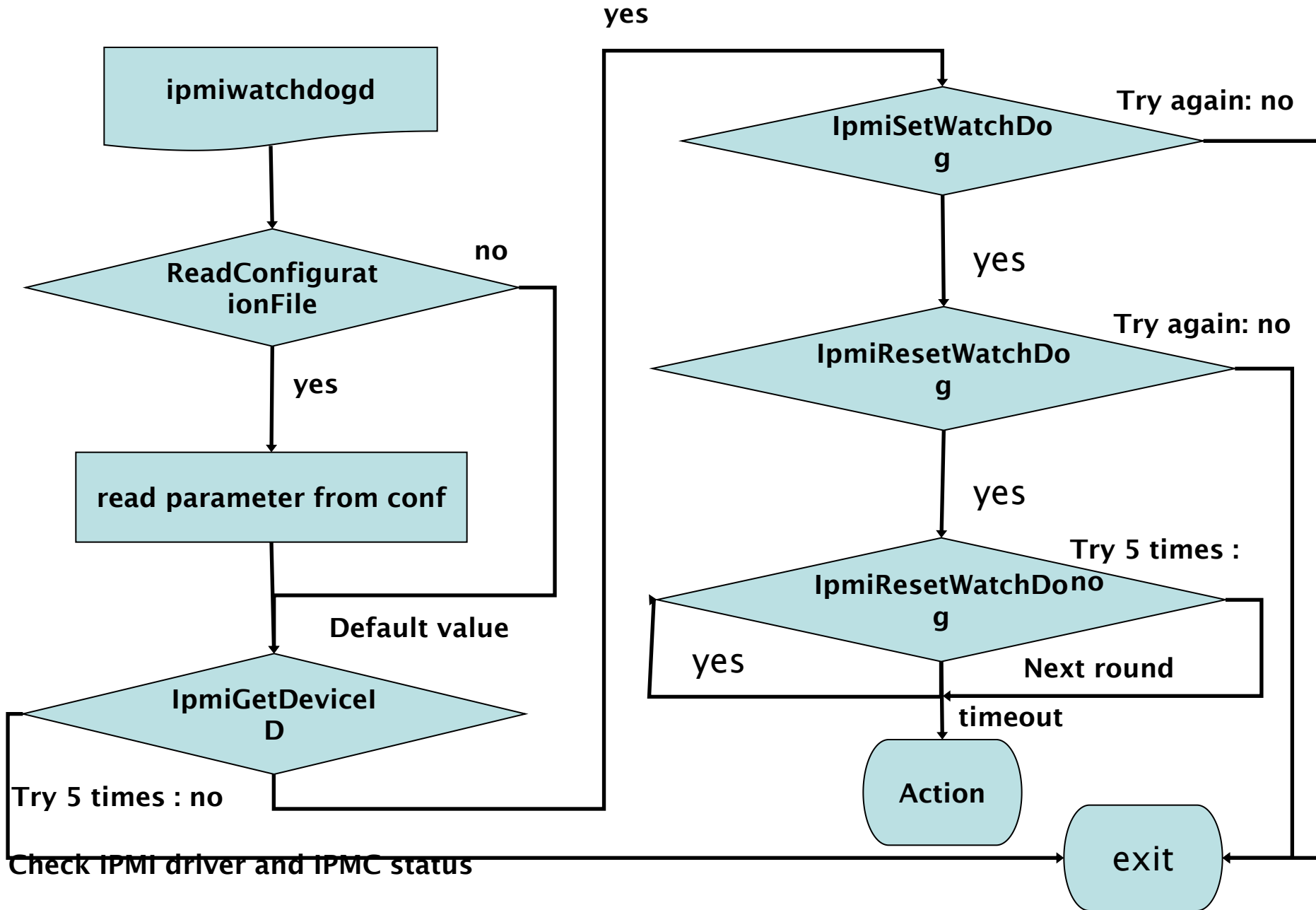
Ipmiwatchdogd layers



Hpmcmd layers



ipmiwatchdogd-flowchat



ipmiwatchdogd-marcos

```
#define DEBUG "debug"

#define ConfigurationFileDir "/etc/ipmiwatchdog.conf"
#define IPMIDeviceName1 "/dev/ipmi0"
#define IPMIDeviceName2 "/dev/ipmidev/0"

#define IPMI_TIMEOUT "Timeout"
#define IPMI_PRETIMEOUT "Pretimeout"
#define IPMI_INTERVAL "Interval"
#define IPMI_PRETIMEOUTINTERRUPT "INT_Pretimeout"
#define IPMI_ACTION "Action"

#define Get_Device_ID 0x01
#define Reset_Watchdog_Timer 0x22
#define Set_Watchdog_Timer 0x24
#define Get_Watchdog_Timer 0x25

#define CONFIG_LINE_LEN 100
```

ipmiwatchdogd-structs

/* Struct for IPMI watchdog parameter */

typedef struct IpmiWatchdogParameter

{

unsigned char TimeoutMsb;

unsigned char TimeoutLsb;

unsigned short PreTimeout;

unsigned short Interval;

enum TimeoutAction

{

NoAction,

HardReset,

PowerDown,

PowerCycle

}TimeoutAction;

enum PreTimeouInterrupt

{

None,

SMI,

NMI,

MSI

}PreTimeouInterrupt ;

}IpmiWatchdogParameter;

Ipmiwatchdog-structs (cont)

struct ipmi_req

```
{  
    unsigned char  *addr;  
    unsigned int   addr_len;  
    long           msgid;  
    struct ipmi_msg msg;  
};
```

struct ipmi_msg

```
{  
    unsigned char netfn;  
    unsigned char cmd;  
    unsigned short data_len;  
    unsigned char *data;  
};
```

struct ipmi_system_interface_addr

```
{  
    int      addr_type;  
    short    channel;  
    unsigned char lun;  
};
```

union IpmiOctToHex

```
{  
    unsigned short TimeOutOct;  
    unsigned char TimeOutHex[2];  
};
```

Ipmiwatchdogd-functions introduction

/* Read the configuration from conf file and set the watchdog parameters struct */
static int ReadConfigurationFile(char *file)

Input value: configuration file directory

Return value: ccode

/* Open related files, mainly used by function SendIpmiCommand*/
int OpenIpmiRelatedFile(char *FileDirectory1, char *FileDirectory2)

Input value: IPMC device file

Return value: FD

/* Close the opened file */
void CloseIpmiRelatedFile(int IpmiRelatedFD)

Input value: FD

Return value: none

/* Used to format the Get Device ID command, create CMD and return the struct ipmi_req */

struct ipmi_req IpmiGetDeviceID(void)

Input value: none

Return value: struct ipmi_req

/* Used to format the Set Watchdog Timer , create CMD and return the struct ipmi_req. */

struct ipmi_req IpmiSetWatchDog(struct IpmiWatchdogParameter Parameter)

Input value: struct IpmiWatchdogParameter Parameter

Ipmiwatchdogd-functions introduction (cont)

```
/* Used to format the Reset IPMI Watchdog , create the request and return the struct ipmi_req. */
```

```
struct ipmi_req IpmiResetWatchDog(void)
```

Input value: none

Return value: struct ipmi_req

```
/* Used to format the Get IPMI Watchdog , create the request and return the struct ipmi_req. */
```

```
struct ipmi_req IpmiGetWatchDog(void)
```

Input value: none

Return value: struct ipmi_req

```
/* Used to Send the RAW IPMI command, require the format struct ipmi_req. */
```

```
/* If Debug Mode opened, the function can return all the response data */
```

```
int SendIpmiCommand(struct ipmi_req IPMICmdReq)
```

Input value: struct ipmi_req

Return value: ccode

```
/*Function used to read the configuration from the conf file in firectory defined by 'filename' */
```

```
static int ReadConfigurationFile(char *file)
```

Input value: directory of the configuration file

Return value: ccode

```
/* Main function of ipmiwatchdogd*/
```

```
int main(int argc, char *argv[])
```


Ipmiwatchdogd-configuration file

The configuration file for ipmiwatchdogd daemon; the conf directory
“/etc/ipmiwatchdogd.conf”

===== Actual Parameters =====

Timeout = 600

Pretimeout = 10

INT_Pretimeout = 0

Interval = 10

Action = 1

=====

Timeout Action - the action will be taken if timer expires

#

TimeoutAction = 0 represents NoAction;

TimeoutAction = 1 represents HardRest;

TimeoutAction = 2 represents PowerDown;

TimeoutAction = 3 represents PowerCycle;

#

Pre-timeout interrupt

INT_PreTimeout = 0 represents None;

INT_PreTimeout = 1 Represents SMI ;

INT_PreTimeout = 2 represents NMI;

INT_PreTimeout = 3 represents MSI;

----- NOTE -----

Ipmiwatchdogd-start/stop/restart script

```
#!/bin/bash
# S99zIPMIWATCHDOGD used to control the ipmiwatchdogd daemon "nore detail in actual scripts"
#the ipmiwatchdogd deamon location /usr/sbin/ipmiwatchdogd
DAEMON=/usr/sbin/ipmiwatchdogd
RetrunValue=0
#Check if the daemon is there.
if [ -f $DAEMON ]
then
    echo "$DAEMON is here"
else
    echo "$DAEMON is not here"
    exit 1
fi
# See how ipmiwatchdogd daemon called.
case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    restart)
        restart
        ;;
    *)
        echo " Usage: ipmiwatchdog start|stop|restart "
        ;;
esac
exit $RetrunValue
```

Usage

1, install the RPM

- 1.0 `#rpm -ivh ipmiwatchdogd-1.0-0.x86_64.rpm`
or `#rpm -ivh ipmiwatchdogd-1.0-0.ppc_e500v2.rpm`
- 1.1 daemon will be installed to directory `"/usr/sbin/ipmiwatchdogd"`;
- 1.2 configuration will be installed to directory `"/etc/ipmiwatchdog.conf"`;
- 1.3 Script will be installed to directory `"/etc/init.d/S99zIPMIWATCHDOGD"`.

2, modify the `"/etc/ipmiwatchdog.conf"`;

by default: timeout "60 seconds"; pre-timeout "10 seconds"; timeout action "hard reset"; pre-timeout interrupt "none"; interval "10 seconds"

3, try to see if the IPMC watchdog works well

`#!/usr/sbin/ipmiwatchdogd debug`

Debug log will be output for 60 times (according to timeout value) ,check if the watchdog set and reset.

You can also use IPMI RAW comamnd or homcmd command

`#hpmcmd -c watchdog get`

4, if the watchdog works well, if not (contact me Sam.Lee@emerson.com :-)

- 4.1 copy the script `"S99zIPMIWATCHDOGD"` to `"/etc/rc3.d/"`
- 4.2 `#!/etc/rc3.d/S99zIPMIWATCHDOGD` restart

5, tips, if there something with the ipmiwatchdogd,

- 5.1 you could stop the daemon `#kill -9 "ipmiwatchdogdmon pid"`
- 5.2 stop watchdog remotely `#ipmicmd -k "0f IPMB 06 24 00 00 00 00 00 00" smi 0`

Enjoy it :-) thank you.