```
#!/bin/bash
__acduroy__
#********************************************************************************************
# Description     : This bash script will fetch, display and monitor the SES enclosure status page 2
#                     information using the Linux sg_ses utility. Any critical condition exist will prompt the
#                     end-user a warning message and it will halt the program.
# Usage            :  ./read_SS8462_ses_new
#***************************************

function select_command_option {
    #*** option to select command to run ***
    while [[ $# -gt 1 ]]
    do
        key="$1"
        #*** Select options -t for temp; -s for status
        case $key in
            #*** Get Enclosure Temp ***
            -t|--temp)
                # echo Display Enclosure Temperature
                SENSOR_NAME="$2"
                DEVICE_NAME="$3"
                read_enclopsure_temp
                shift  #pass argument or value
                ;;
            #*** Get Enclosure Overall Status ***
            -s|--status)
                # echo Display Enclosure Overall Status
                DEVICE_NAME="$2"
                EXPECTED_VALUE="$3"
                read_enclosure_page2
                shift  #pass argument or value
                ;;
            *)
         ;;
        esac
        shift #pass argument or value
    done
    echo element# = "${SENSOR_NAME}"
}

function select_option {
    OPTIND=1
    while getopts "ts:" opt
    do
        case "$opt" in
        t)
            SENSOR_NAME="$3"
            DEVICE_NAME="$2"
            ;;
        s)
            DEVICE_NAME="$2"
            EXPECTED_VALUE="$3"
            ;;
        esac
    done
    shift $((OPTIND-1))
}

function read_enclosure_temp {
#* Usage: ./read_ss8460_temp -g element_type -o overtemp_setpoint -u undertemp_setpoint
#* Description: To get the temp of the enclosure and to set the over/under temp trigger
#* Date: 02-16-2017
#* Rev. 1

    echo "*Read enclosure temp and over-under temp status*"
    #*** Loop forever ***
    for ((;;))
    do
        #*** Variable Initialization ***
        temp=0
        element_status=0
        smart=0
        date
        #*** Selection of temp sensor elemet type ***
        #*** Get Overall Enclosure Temp and Status ***
        #temp[0]=$(sg_ses -p 0x02 -H -s /dev/$DEVICE_NAME |head -n 28 |tail -n 1 |cut -c 52-54)
        #element_status[0]=$(sg_ses -p 0x02 -H -s /dev/$DEVICE_NAME |head -n 28 |tail -n 1 |cut -c 55-
56)
        #*** Get Sensor 2 Temp and Status ***
        #temp[2]=$(sg_ses -p 0x02 -H -s /dev/$DEVICE_NAME |head -n 29 |tail -n 1|cut -c 26-28)
        #element_status[2]=$(sg_ses -p 0x02 -H -s /dev/$DEVICE_NAME |head -n 29 |tail -n 1|cut -c 29-
31)
        #*** Get Sensor 6 Temp and Status ***
        #temp[6]=$(sg_ses -p 0x02 -H -s /dev/$DEVICE_NAME |head -n 30 |tail -n 1|cut -c 26-28)
        #element_status[6]=$(sg_ses -p 0x02 -H -s /dev/$DEVICE_NAME |head -n 30 |tail -n 1 |cut -c 29-
31)
        #*** Get Sensor 7 Temp and Status ***
        #temp[7]=$(sg_ses -p 0x02 -H -s /dev/$DEVICE_NAME |head -n 30 |tail -n 1 |cut -c 35-37)
        #element_status[7]=$(sg_ses -p 0x02 -H -s /dev/$DEVICE_NAME |head -n 30 |tail -n 1 |cut -c 38-
40)

        smart[0]=$(smartctl -a /dev/sda |head -n 6 |tail -n 1 |cut -c 18-33)
        smart[1]=$(smartctl -a /dev/sda |head -n 69 |tail -n 1 |cut -c 38-40)
        #*** check for exit status ***
        if [ $? -eq 0 ]
        then
```

```
            #*** Display Enclosure Temperature ***
            #echo Overall Enclosure Temperature = ${temp[0]}  Status = ${element_status[0]}

            echo model no=${smart[0]}
            echo hdd temp=${smart[1]}
        else
            echo command error !!!
            exit 1

        fi

    done

#***** End of Function read_enclosure_temp *****
}


function read_enclosure_page2 {
    echo "*Read enclosure page 2*"
    var1=$2
    for ((;;))
    do
        date
        #*** preserve last reading ***
        var2=$var1
        #*** get the enclosure status ***
        var1=$(sg_ses -p 0x02 -H /dev/$1 |head -n 4|tail -n 1|cut -c 11-13)
        #*** check for exit status ***
        if [ $? -eq 0 ]
        then
            #****** compare to last reading ********
            if [ $var1 != $var2 ]
            then
                echo *****status changed******
                echo  current status="$var1", last="$var2"
                break
            fi
        else
            break
        fi
    done
}

function check {
    echo "checking function !!!"
}
```

```
function options_main {
    echo "Choose what enclosure's page  to read !!!"

    OPTIONS=("OverallStatus"  "DisplaySMARTinfo"  "Quit")

    select opt in "${OPTIONS[@]}"
    do
         case "$opt" in
            "OverallStatus")
               select_command_option
               read_enclosure_page2
               echo "I don't know what is happening here… !!!"
               ;;

           "DisplaySMARTinfo")
               #select_command_option
               read_enclosure_temp
               ;;

           "Quit")
               echo "Thanks you for using this program !!! bye . bye..."
                break
               ;;

           *)echo invalid option;;
        esac
    done
}


function main {
    #select_command_option
    options_main
    #read_enclosure_temp
    #exit

}

#***** call main function *****
main
```