

```

#!/usr/bin/perl -w
#
# fah_log_parser.pl
#
# Collect Folding@Home logs (log.txt) from local machines
# Write new data to history files
# Run from cron once daily (or manually, or whenever ;- )
#
# REQUIRES: log_collector (see below)
#
# CC-BY-SA 2007
# Benjamin Gayle N1NP
#
# Re-Write for Folding@Home 7.x.x series clients
# 11 December 2015

use strict;
use utf8;

# Main log storage directory, change to suit your location:
my $log_dir = "$ENV{HOME}/fah/logs";
opendir(my $LOG_DIR,$log_dir) or die "Cannot open $log_dir: $! \n";

# Get the target info from log_collector (separate file, create your own).
# log_collector stores the shell commands for retrieving each logfile, e.g. by scp
# If you use scp, you will need to run ssh-keygen to get a key pair, then put the
# public key into $home/.ssh/authorized keys on each target machine.
# EXAMPLE line (figure out your own local method for moving the files):
# scp -i $HOME/.ssh/id_rsa 192.168.42.43:$HOME/fah/7-4-4/log.txt $HOME/fah/logs/veronika.log
# That means: scp -i {key location} {file to get} {where to put it}

my $targets_file_name = "log_collector";
open(my $TARGETS_FH,$targets_file_name)
    or die "Cannot open $targets_file_name: $! \n";
my @targets = <$TARGETS_FH>;
chomp @targets;
close $TARGETS_FH;

# Retrieve the files (run the shell commands found in log_collector)
foreach my $target (@targets) {
    system($target) == 0
        or die "Could not retrieve log: error $? \n";
}

# Get the current system time and date:
my($sec,$min,$hour,$mday,$mon,$year,$yday,$isdst) = localtime(time);
$year += 1900;
$mon += 1;
my $today = $mday . "-" . $mon . "-" . $year;

# Get the list of log files to process:
my @log_files = grep { /\.log$/ && -f "$log_dir/$_" } readdir($LOG_DIR);
chomp @log_files;

foreach my $log_file_name (@log_files) {
    my @results;

```

```

my $header = $log_file_name;
substr($header,-3) = q{};
my $history_file_name = $log_dir . "/" . $header . "history";

open(my $LOG_FH,$log_file_name) or die "Cannot open $log_file_name: $! \n";
my @projects_raw = grep { /Project/ || /credit/ || /WARNING/ || /:ERROR/ } <$LOG_FH>;
# "\e[93m . . . \e[0m" are for color YELLOW!
# "\e[91m . . . \e[0m" are for color RED!
chomp @projects_raw;

if (stat $history_file_name) {
    open(my $HISTCMP_FH, $history_file_name)
        or die "Could not open $history_file_name: $! \n";
    foreach my $candidate (@projects_raw) {
        seek $HISTCMP_FH,0,0; # Force the pointer to the beginning of the file
        # Must escape all the metachars ()\|[]{}^$*+?. that may show up in the string!
        my $meta_candidate = quotemeta $candidate;
        if ( grep { /$meta_candidate/ } <$HISTCMP_FH> ) {
            # Already logged, skip
        }
        else {
            push @results, $candidate;
            # NEW result, add it to the list
        }
    }
    close $HISTCMP_FH;
}
else { @results = @projects_raw; } # Only for a fresh start with no .history file

# Save the data to the .history file like this:
# date, wu start time UTC, project info
# 12-12-2015:08:57:45:WU00:FS00:0xa4:Project: 6381 (Run 2, Clone 71, Gen 132)
# 12-12-2015:16:31:56:WU00:FS00:Final credit estimate, 5994.00 points
open(my $HISTORY_FH, ">>$history_file_name")
    or die "Could not open $history_file_name: $! \n";
foreach my $result (@results) {
    print {$HISTORY_FH} $today, ":", $result, "\n";
}
close $HISTORY_FH;
close $LOG_FH;
}

close $LOG_DIR;
# END MAIN

```