```perl
#!/bin/perl

use strict;
use warnings;
use HTML::Entities;

my ( $R, $G, $Y, $B, $P, $C, $W );
$R="\033[31m";$G="\033[32m";$Y="\033[33m";$B="\033[34m";$P="\033[35m";$C="\033[36m";$W="\033…

my $wdir = "/tmp/henlo";
`mkdir $wdir` unless -e $wdir;
chdir( $wdir );
my @threads = ();

for ( my $i=0; $i<2; $i++ )
{
        my @label = ( "${Y}current$W", "${Y}target$W" );
        while (1)
        {       print "$W>: $label[$i] thread?\n<: $Y";
                $threads[$i] = <> ;
                chomp $threads[$i];
                last if `curl -s -I "https://alogs.space/robowaifu/res/${threads[$i]}.html"`
                                =~ /^HTTP.*200/ ;
                printf( "$W$W>: thread $Y%s$W not found\n", $threads[$i] );
        }
}
print "$W>: downloading thread $Y${threads[0]}$W\n";
$_ = `curl -# "https://alogs.space/robowaifu/res/${threads[0]}.html"`;
        tr/\n/\001/;
        s/postCell/\n\npostCell/g;
my $text = $_;
my @posts = ();
while (1)
{
        my $loop = 0;
        print "$W>: posts to move? (deliminated by any non-numbers)\n<: $R";
        $_ = <>;
        chomp();
        s/\D+/ /g;
        @posts = split( ' ', $_ );
        foreach my $post (@posts)
        {
                if ( $text !~ /id="$post"/ )
                {
                        print "$W$W>: $R$post$W not found in thread $Y$threads[0]$W\n";
                        $loop = 1;
                }
        }
        last unless $loop || ! scalar @posts;
}
@posts = sort { $a <=> $b } @posts;

###### COLLECT POSTS #########
my @list = ( );
foreach my $post (@posts)
{
        my ( $name, $files, $mssg, @flinks, @fnames ) =
                $text =~ m:id="$post".*?(linkName.*?</a>).*?(panelUploads.*?)?(divMessage.*?…
        $name =~ />([^<]+)/;
        $name = $1;
        $mssg =~ m:>(.*?)</div>: ;
        $_ = $1;
                tr/\001/\n/;
                s:<span class="(pinkText|greenText)">(.*?)</span>:$2:gi;#<|>
                s|<a.*?quote.*?>(.*?\d*)</a>|$1|gi;                      #>>
                s/<span class="spoiler">(.*?)<\/span>/\*$1\*/g;          #**
                s/<span class="redText">(.*?)<\/span>/==$1==/g;          #==
```

```
                s/<em>(.*?)<\/em>/''$1''/g;                                     #''
                s/<u>(.*?)<\/u>/__$1__/g;                                       #__
                s:<s>(.*?)</s>:~~$1~~:g;                                        #~~
                s|<span class="aa"(.*?)</span>|[aa]${1}[/aa]|g;                 #[aa]
                s:<code>(.*?)</code>:[code]${1}[/code]:g;                       #[code]
                s:<a href=".*>(.*?)</a>::g;                                     #url

        $mssg = decode_entities( $_ );
        while ( $files && $files =~ m:nameLink.*?(/.media/[^"]+).*?download="([^"]+):gc )
        {
                push @flinks, $1;
                $_ = decode_entities( $2 );
                tr/ /_/;
                push @fnames, $_;
        }
        push @list, $name, $mssg, (join " ", @flinks), (join " ", @fnames) ;
}

####### CHECK AND CONFIRM #######
for ( my $i=0; $i<@posts; $i++ )
{
        printf( "$W>: [%d/%d]\n", $i+1, scalar @posts );
        printf( "$W>: POST: \t$R%s\n", $posts[$i] );
        printf( "$W>: NAME: \t$C%s\n", $list[ $i * 4 + 0 ] );
        printf( "$W>: TEXT: \t$G%s\n", $list[ $i * 4 + 1 ] );
        printf( "$W>: FLINKS: \t$B%s\n", $list[ $i * 4 + 2 ] );
        printf( "$W>: FNAMES: \t$P%s\n\n", $list[ $i * 4 + 3 ] );
}
printf( "$W>: confirm moving $R%s$W in $Y%s$W to $Y%s$W? [y/n]\n<: ",
                                join( ' ', @posts ) , $threads[0], $threads[1]  );
die "$W>: k, fix it then\n" if ( <> !~ /^(y|yes)$/i );

####### POST IN NEW THREAD #######
my @newposts = ();
for ( my $i=0; $i<@posts; $i++ )
{
        printf( "$W>: [%d/%d] processing $R%s$W\n", $i + 1, scalar @posts, $posts[$i] );

        ## FIX QUOTES ##
        my $mssg = $list[ $i * 4 + 1 ];
        for ( my $j=0; $j<@posts; $j++ )
        {       print "$W>: $R$posts[$i]$W quoting $R$posts[$j]$W changed to $R${newposts[$j…
                        if ( $mssg =~ s/>>$posts[$j]/>>$newposts[$j]/g ) ;
        }
        ## DOWNLOAD FILES ##
        my @flinks = split(' ', $list[ $i * 4 + 2 ]);
        my @fnames = split(' ', $list[ $i * 4 + 3 ]);
        for ( my $j=0; $j<@flinks; $j++ )
        {
                print "$W>: downloading: $P$fnames[$j]\n" ;
                if ( `curl -I -# \"https://alogs.space$flinks[$j]\"` =~ /^HTTP.*200/ )
                {       `curl -s \"https://alogs.space$flinks[$j]\" -o \"$wdir/$fnames[$j]\"…
                }
                else
                {       print "$R>: file '$P$fnames[$j]$W' not found, retry? [y/n]\n" ;
                        $j-- if ( <> =~ /^(y|yes)$/i );
                }
        }
        ## POST ##
        my $curlcmd =   "curl -i -# -X POST "
                                ."'https://alogs.space/replyThread.js' "
                                ."-b \"$wdir/cookie\" "
                                ."--form-string \"message=$mssg\" "
                                ."-F \"email=sage\" "
                                ."-F \"name=${list[ $i * 4 + 0 ]}\" "
                                ."-F \"password=test\" "
                                ."-F \"boardUri=robowaifu\" "
```

```
                                ."-F \"threadId=${threads[1]}\" ";
        # add files
        foreach my $file (@fnames)
        {
                $curlcmd = $curlcmd . " -F \"files=\@$wdir/$file;\" " ;
        }

        $_ = `$curlcmd`;
        if ( /blockBypass/ )
        {
                while ( 1 )
                {
                        #captcha key
                        print "$W>: captcha required\n$W";
                        $_ = `curl -i -# 'https://alogs.space/noCookieCaptcha.js'`;
                        my $captchaId = $1 if /name=\s*"captchaId".*value=\s*"([^"]+)/;
                        my $captchaimg = $1 if /id=\s*"imageCaptcha".*src="([^"]+)/;
                        `curl -s "https://alogs.space$captchaimg" -o "$wdir/captcha.jpeg"`;
                        print "$W>: captcha answer?\n<: ";
                        `open "$wdir/captcha.jpeg"&`;
                        my $answer = <>;
                        $curlcmd = "curl -i -# -X POST "
                                        ."'https://alogs.space/solveCaptcha.js' "
                                        ."-F \"captchaId=$captchaId\" "
                                        ."-F \"answer=$answer\"";
                        $_ = `$curlcmd`;
                        if ( /Captcha.solved/i )
                        {
                                #bypass
                                print "$W>: captcha passed\n";
                                $curlcmd = "curl -i -# -X POST "
                                                ."'https://alogs.space/renewBypass.js' "
                                                ."-c \"$wdir/cookie\" "
                                                ."-F \"captcha=$captchaId\" " ;
                                $_ = `$curlcmd`;
                                if ( /Block.bypass.renewed/i )
                                {       print "$W>: bypass aqcuired\n";
                                        $i--; last;
                                }
                                print "$W>: bypass failed\n{$G$_}\n"
                        }
                        else
                        {       print "$W>: captcha failed\n";
                        }
                }
        }
        elsif ( /^location.*html#(\d+)/ )
        {
                push @newposts, $1;
                print "$W>: migrated $R${posts[$i]}$W succesfully to $Y${threads[1]}$W:$R$1$…
                last;
        }
        elsif ( /id=\s*"errorLabel"(.*)</ )
        {       print "$W>: you done goofed ERROR:${G}{ ${1} }${W}\ntry again? [y/n]\n";
                die if ( <> !~ /^(y|yes)$/i );
        }
}
print "$W>: done\n";
####### DELETE FROM OLD THREAD #######
print "$W>: delete { $R@posts$W } from $Y$threads[0]$W?\n<: ";
die "$W>: posts not deleted\n" unless ( <> =~ /^(y|yes)$/i );
foreach my $post (@posts)
{
        print "$W>: deleting $R$post$W\n";
        my $curlcmd =   "curl -i -# -X POST "
                        ."'https://alogs.spac7e/replyThread.js' "
                                ."-b \"$wdir/cookie\" "
```

```
                                ."-F \"robowaifu-${threads[0]}-$post=on\""
                                ."-F \"password=????\" "        # FIXME
                                ."-F \"action=delete\" "
                                ."-F \"reasonReport=what\" ";
}
```