Algorithme du programme principal

Voici l'algorithme du programme principal. Ce dernier est donné dans les détails :

```
Déclaration des modules externe : Débogage et date en jour

### Détection du système d'exploitation ###
Si Système est Windows
        Alors utiliser la commande cls (nettoyer l'écran sous dos)
Fin si
Sinon
        Utiliser la commande clear (nettoyer l'écran sous Linux)
Fin sinon
####################################

####### Déclarations des variables #######
Déclaration des variables
Déclaration de l'expression régulière Apache
Déclaration de l'expression régulière IIS
Déclaration des hash
####################################

########### Test fichier et date ########
Si l'utilisateur entre moins de 3 arguments
        Alors afficher « utilisation script.pl <logfile> <date début> <date fin>
                afficher « Exemple d'utilisation »
Fin si

Stockage des arguments dans des variables

Si l'argument 1 (fichier) n'existe pas
        Alors afficher erreur
Fin si
Si l'argument 2 (date début) est bien de type date
        Alors stocker le jour, le mois et l'année « début » dans 3 variables
Sinon
        Afficher « erreur dans la date de début »
Fin sinon

Si l'argument 3 (date fin) est bien de type date
        Alors stocker le jour, le mois et l'année « fin » dans 3 variables
Sinon
        Afficher « erreur dans la date de fin »
```

Fin sinon
####################################

####### Main #######
Ouverture du fichier log sinon afficher une erreur d'ouverture
Parcourir le fichier log tant qu'il y a des lignes
    Si la ligne est de la forme Apache
        On stock chaque champ dans des variables (IP, jour, mois, année, méthode...)
    Fin si
    Sinon la ligne est de forme IIS
        On stock chaque champ dans des variables (IP, jour, mois, année, méthode...)
    Fin sinon
    Convertir la date du fichier log dans un bon format xx/xx/xxxx
    Convertir la date du fichier log en jour
    Convertir la date de début en jour
    Convertir la date de fin en jour
    Si la date du fichier log se trouve dans la fourchette de date début et date fin
        Alors on compte les IP
            on compte les Referer
            on compte les user agents
            ....
            on stock la somme des kilo-octets sur tout le fichier
            on stock la somme des kilo-octets par IP

    Fin Si

Appel de la fonction d'affichage du menu
Demande du choix à l'utilisateur
Si le choix est 1
    Alors lancer la fonction Nombre d'adresse IP
Fin si
Sinon si le choix est 2
    Alors lancer la fonction Hits IP
Fin sinon si
Sinon si le choix est 3
    Alors lancer la fonction Page
Fin sinon si
Sinon si le choix est 4
    Alors lancer la fonction Referer
Fin sinon si
Sinon si le choix est 5
    Alors lancer la fonction User Agents
Fin sinon si
Sinon si le choix est 6
    Alors lancer la fonction Ko total pour toutes les IP
Fin sinon si
Sinon si le choix est 7
    Alors lancer la fonction Ko total par IP
Fin sinon si

Sinon si le choix est 8
    Alors quitter le programme
Fin sinon si

Sinon le choix est autre
    Afficher que l'utilisateur a saisie une mauvaise option
    Quitter le programme
Fin sinon

Fermeture du fichier
####################

7.1 Algorithme des fonctions

Voici l'algorithme des fonctions. Ce dernier est donné dans les détails :

Fonction affichage
    Nettoyage de l'écran
    Affichage de phrase de bienvenue
    Affichage du menu
Fin fonction affichage

Fonction Nombre IP
    Nettoyage de l'écran
    Ouverture en incrémentiel ou Création du fichier log_nbr_IP
    Ecriture dans le fichier log_nbr_IP la date début – date fin
    Initialiser une variable compteur
    Pour chaque IP
        Mettre dans l'ordre suivant le nombre de fois qu'elle apparaît
        Incrémenter la variable compteur
    Afficher à l'écran le résultat
    Ecriture du résultat dans le fichier log_nbr_IP
    Afficher « Ecriture en cours... »
    Fonction mess()
Fin fonction Nombre IP

Fonction IP
    Nettoyage de l'écran
    Ouverture en incrémentiel ou Création du fichier log_IP
    Ecriture dans le fichier log_IP la date début – date fin
    Pour chaque IP
        Mettre dans l'ordre suivant le nombre de fois qu'elle apparaît
        Afficher à l'écran le résultat (seulement 10 IP)
        Ecriture du résultat dans le fichier log_IP
    Afficher « Ecriture en cours... »
    Fonction mess()
Fin fonction IP

Fonction page
Nettoyage de l'écran
Ouverture en incrémentiel ou Création du fichier log_page
Ecriture dans le fichier log_page la date début – date fin
Pour chaque page
Mettre dans l'ordre suivant le nombre de fois qu'elle apparaît
Afficher à l'écran le résultat (seulement 10 pages)
Ecriture du résultat dans le fichier log_page
Afficher « Ecriture en cours... »
Fonction mess()
Fin fonction page

Fonction referer
Nettoyage de l'écran
Ouverture en incrémentiel ou Création du fichier log_referer
Ecriture dans le fichier log_referer la date début – date fin
Pour chaque referer
Mettre dans l'ordre suivant le nombre de fois qu'il apparait
Afficher à l'écran le résultat (seulement 10 referers)
Ecriture du résultat dans le fichier log_referer

Afficher « Ecriture en cours... »
Fonction mess()
Fin fonction referer

Fonction User Agents
Nettoyage de l'écran
Ouverture en incrémentiel ou Création du fichier log_ua
Ecriture dans le fichier log_ua la date début – date fin
Pour chaque user agents
Mettre dans l'ordre suivant le nombre de fois qu'il apparait
Afficher à l'écran le résultat
Ecriture du résultat dans le fichier log_ua
Afficher « Ecriture en cours... »
Fonction mess()
Fin fonction User Agents

Fonction ko
Nettoyage de l'écran
Ouverture en incrémentiel ou Création du fichier log_ko
Ecriture dans le fichier log_ko la date début – date fin
Récupérer la variable du programme principal
Afficher à l'écran le résultat
Ecriture du résultat dans le fichier log_ko
Afficher « Ecriture en cours... »
Fonction mess()
Fin fonction ko

Fonction koip
    Nettoyage de l'écran
    Ouverture en incrémentiel ou Création du fichier log_ko_IP
    Ecriture dans le fichier log_ko_IP la date début – date fin
    Pour chaque IP
        Mettre dans l'ordre suivant le nombre de ko
        Afficher à l'écran le résultat (seulement 10 IP)
        Ecriture du résultat dans le fichier log_ko_IP
    Afficher « Ecriture en cours... »
    Fonction mess()
Fin fonction koip

Fonction mess
    Patienter 2 secondes
    Afficher « Ecriture réussie »
    Afficher « A bientôt »
Fin fonction mess

Programme final – Perl

```
#!/usr/bin/perl -w
use strict;
use Date::Calc qw( Date_to_Days );

####### DETECTION SYSTEME #####
my $screen_clear;
if ($^O =~ /MSWin32/) {
   $screen_clear = "cls";
}
else {
   $screen_clear = "clear";
}
################################

########## VARIABLES ##########

my $APACHE = q{([\d.]+)[-\s]+\[(\w+)/(\w+)/(\d+)[\d:]+\s\+\d+\] "([A-Z]+) /(\S+) .*" (\d+)
(\d+) "(.*)" "(.*)"};
my $IIS = q{(\d+)-(\d+)-(\d+) [\d:]+ ([\d.]+) - ([A-Z]+) /(.*) (\d+) (\d+) - (.*)
((?:http://)?www.*)/};
my ($choix, $sombyte);
my ($ip,$jour,$mois,$annee,$method,$page,$ret_code,$byte,$referer,$user_a);
my ($lower,$upper,$date);
my ($jourin,$jourout,$moisin,$moisout,$anneein,$anneeout);
my (%haship,%hashpage,%hashref,%hashua,%hashko);
################################

#### TEST FICHIER ET DATE #####

if ($#ARGV != 2){
        print ("\nUtilisation : script.pl <fichier_log> <date_debut> <date_fin>\n");
        print ("Exemple : script.pl log.txt 20/01/2009 20/02/2009 \n");
        exit();
}

my $file = $ARGV[0];
my $datein = $ARGV[1];
my $dateout = $ARGV[2];

if (! -e $file){
        print $file," existe pas.\n";
        exit();
}
```

```
if ($datein =~ m{^(\d+)/(\d+)/(\d+)$}) {

	$jourin = $1;
	$moisin = $2;
	$anneein = $3;
}

else {
	print ("Erreur de saisie sur date_debut\n");
	exit();
}

if ($dateout =~ m{(\d+)/(\d+)/(\d+)}) {
	$jourout = $1;
	$moisout = $2;
	$anneeout = $3;
}

else {
	print ("Erreur de saisie sur date_fin\n");
	exit();
}
################################

############ MAIN #############
open (Fichier, "< $file") or die "Ne peux pas ouvrir le fichier $ARGV[0]\n";

while (my $ligne = <Fichier>){
		if ($ligne =~ $APACHE){

	($ip,$jour,$mois,$annee,$method,$page,$ret_code,$byte,$referer,$user_a) = ($1, $2,
$3, $4, $5, $6, $7, $8, $9, $10);
	}
		elsif ($ligne =~ $IIS){

	($annee,$mois,$jour,$ip,$method,$page,$ret_code,$byte,$user_a,$referer) = ($1, $2,
$3, $4, $5, $6, $7, $8, $9, $10);
	}
		$mois = '01' if ($mois eq "Jan");
		$mois = '02' if ($mois eq "Feb");
		$mois = '03' if ($mois eq "Mar");
		$mois = '04' if ($mois eq "May");
		$mois = '05' if ($mois eq "Apr");
		$mois = '06' if ($mois eq "Jun");
		$mois = '07' if ($mois eq "Jul");
		$mois = '08' if ($mois eq "Aug");
		$mois = '09' if ($mois eq "Sep");
		$mois = '10' if ($mois eq "Oct");
		$mois = '11' if ($mois eq "Nov");
		$mois = '12' if ($mois eq "Dec");
```

```
                $lower = Date_to_Days($anneein,$moisin,$jourin);
                $upper = Date_to_Days($anneeout,$moisout,$jourout);
                $date = Date_to_Days($annee,$mois,$jour);
                if (($date >= $lower) && ($date <= $upper)) {
                        $haship{$ip} += 1;
                        $hashpage{$page} += 1;
                        $hashref{$referer} += 1;
                        $hashua{$user_a} += 1;
                        $hashko{$ip} += $byte;
                        $sombyte += $byte;
                }
}

&subaffich();
$choix = (<STDIN>);

if ($choix == 1) {
        &subnbip();
}

elsif ($choix == 2) {
        &subip();
}

elsif ($choix == 3) {
        &subpage();
}

elsif ($choix == 4) {
        &subref();
}

elsif ($choix == 5) {
        &subua();
}
elsif ($choix == 6) {
        &subko();
}
elsif ($choix == 7) {
        &subkoip();
}
elsif ($choix == 8) {
   print "** Au revoir **\n";
        exit();
}
else {
   print "\nVous avez saisie un mauvais choix\n\n";
   print "** Au revoir **\n";
   exit();
}
```

```
close(Fichier);
################################

############# SUB #############
sub subaffich {
        system("$screen_clear");
        print " |      Bienvenu dans le programme d\'analyseur de log       |\n";
        print " |              Programmer par Nickname                      |\n";
        print " |-----------------------------------------------------------|\n\n";
        print " Veuillez entrer le numero correspondant a l\'action voulue :\n\n";
        print "    1.Nombre d'adresses IP differentes\n";
        print "    2.Hits des 10 adresses IP les plus presentes\n";
        print "    3.Hits des 10 pages les plus visitees\n";
        print "    4.Hits des 10 premieres sources - URLs precedentes\n";
        print "    5.Hits des user agent les plus presents\n";
        print "    6.Somme en ko pour toutes les IP\n";
        print "    7.Somme en ko par IP - 10 Resultats\n";
        print "    8.Quitter le programme\n";
}

sub subnbip {
        system("$screen_clear");
        open(SORTIE, ">>log-nbre-ip.txt");
        print SORTIE "\n*** Du $jourin/$moisin/$anneein au $jourout/$moisout/$anneeout
***\n\n";
        my $i = 0;
        foreach $ip ( sort { $haship{$b} <=> $haship{$a} } keys %haship) {
                $i++;
                }
        print "\nNombre d'adresses IP differentes dans le fichier $file : $i IP\n";
        print SORTIE "Nombre d'adresses IP differentes dans le fichier $file : $i IP\n";
        mess();
}

sub subip {
        system("$screen_clear");
        open(SORTIE, ">>log-ip.txt");
        print SORTIE "\n*** Du $jourin/$moisin/$anneein au $jourout/$moisout/$anneeout
***\n\n";
        my $i = 0;
        foreach $ip ( sort { $haship{$b} <=> $haship{$a} } keys %haship) {
                print "IP : $ip a ete rencontre $haship{$ip} fois\n";
                print SORTIE "IP : $ip a ete rencontre $haship{$ip} fois\n";
                $i++;
                last if ($i == 10);
                }
        print "\nEcriture dans le fichier log-ip.txt en cours...\n";
        mess();
}
```

```
sub subpage {
	system("$screen_clear");
	open(SORTIE, ">>log-page.txt");
	print SORTIE "\n*** Du $jourin/$moisin/$anneein au $jourout/$moisout/$anneeout
***\n\n";
	my $i = 0;
	foreach $page ( sort { $hashpage{$b} <=> $hashpage{$a} } keys %hashpage) {
		print "Page : $page a ete rencontre $hashpage{$page} fois\n";
		print SORTIE "Page : $page a ete rencontre $hashpage{$page} fois\n";
		$i++;
		last if ($i == 10);
		}
	print "\nEcriture dans le fichier log-page.txt en cours...\n";
	mess();
}

sub subref {
	system("$screen_clear");
	open(SORTIE, ">>log-referer.txt");
	print SORTIE "\n*** Du $jourin/$moisin/$anneein au $jourout/$moisout/$anneeout
***\n\n";
	my $i = 0;
	foreach $referer ( sort { $hashref{$b} <=> $hashref{$a} } keys %hashref) {
		print "Referer : $referer a ete rencontre $hashref{$referer} fois\n";
		print SORTIE "Referer : $referer a ete rencontre $hashref{$referer} fois\n";
		$i++;
		last if ($i == 10);
		}
	print "\nEcriture dans le fichier log-referer.txt en cours...\n";
	mess();
}

sub subua {
	system("$screen_clear");
	open(SORTIE, ">>log-user-agent.txt");
	print SORTIE "\n*** Du $jourin/$moisin/$anneein au $jourout/$moisout/$anneeout
***\n\n";
	foreach $user_a ( sort { $hashua{$b} <=> $hashua{$a} } keys %hashua) {
		print "User agent : $user_a a ete rencontre $hashua{$user_a} fois\n";
		print SORTIE "User agent : $user_a a ete rencontre $hashua{$user_a} fois\n";
		}
	print "\nEcriture dans le fichier log-user-agent.txt en cours...\n";
	mess();
}
```

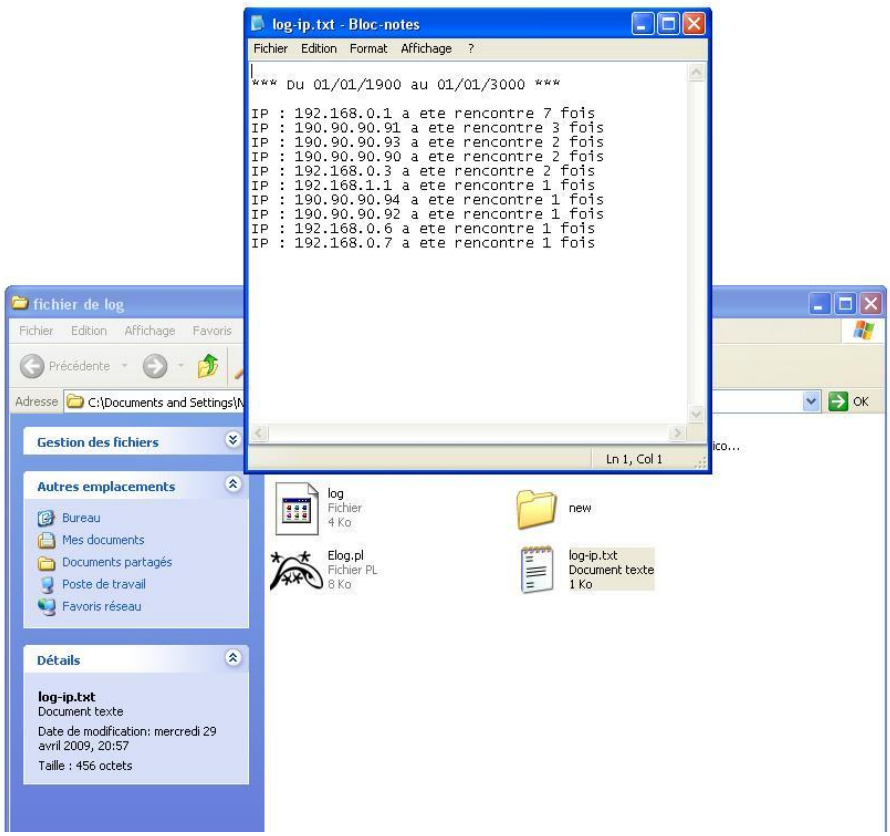
```
sub subko {
        system("$screen_clear");
        open(SORTIE, ">>log-ko.txt");
        print SORTIE "\n*** Du $jourin/$moisin/$anneein au $jourout/$moisout/$anneeout
***\n\n";
        print "Somme des Kilo-octets du fichier log : $sombyte ko\n";
        print SORTIE "Somme Kilo-octets : $sombyte\n";
        print "\nEcriture dans le fichier log-ko.txt en cours...\n";
        mess();
}

sub subkoip {
        system("$screen_clear");
        open(SORTIE, ">>log-ko-ip.txt");
        print SORTIE "\n*** Du $jourin/$moisin/$anneein au $jourout/$moisout/$anneeout
***\n\n";
        my $i = 0;
        foreach $ip ( sort { $hashko{$b} <=> $hashko{$a} } keys %hashko) {
                print "IP : $ip a utiliser $hashko{$ip} ko\n";
                print SORTIE "IP :$ip a utilise $hashko{$ip} ko\n";
                $i++;
                last if ($i == 10);
                }
        print "\nEcriture dans le fichier log-ko-ip.txt en cours...\n";
        mess();
}

sub mess {
        sleep(1);
        print "\nEcriture reussie\n";
        print "\n ** A Bientot **\n";
        exit;
}
##############################
```

Capture d'écran

Vérification dans le fichier log-ip.txt :



Le programme a été testé sur Windows XP SP3 et sur une distribution Linux Backtrack v4.0.