

Pierre Pompidor - vendredi 15 janvier 2009 - 8h30-10h30

## Compréhension d'un script système écrit en Perl : 11 points

Pour expliquer ce que fait le script suivant qui parcourt une arborescence de répertoires, **décrivez les données sur lesquelles ce script va fonctionner, la façon dont il est appelé et son résultat.** Par ailleurs, **expliquez soigneusement le rôle des cinq expressions régulières utilisées.**

*Ce que je veux savoir, c'est ce que vous avez compris des fonctionnalités du script, et non pas que for est une instruction de boucle, ou que if est celle d'un test...*

```
#!/usr/bin/env perl
sub parcours {
    my $r = shift; my $e = shift; my $f = shift; # $r, $e et $f : paramètres de la fonction
    if (opendir DIR, $r) { # Ouverture d'un répertoire dont le nom est contenu dans $r
        my @liste = readdir DIR; # Lecture du répertoire
        foreach $fichier (@liste) {
            if ($fichier !~ /\.\./) { # !~ signifie not =~
                # (teste si le motif ne se trouve pas dans la chaîne)
                if (-d "$r/$fichier") { # -d teste si ce qui suit est un répertoire
                    parcours("$r/$fichier", $e, $f); # la fonction est appelée sur le répertoire
                }
            } else {
                if ($fichier =~ /^f$/)
                {
                    $i = 0;
                    if (open FD, "$r/$fichier") {
                        @lignes = <FD>;
                        foreach $ligne (@lignes) { if ($ligne =~ /$e/) { $i++; } }
                        close FD;
                        if ($i > 0) {
                            # Attention %resultats est un tableau à deux dimensions
                            if ($resultats{$i}) {
                                if ($resultats{$i}{$r}) { $resultats{$i}{$r} .= ":$fichier"; }
                                else
                                    { $resultats{$i}{$r} = $fichier; }
                            }
                            else { $resultats{$i}{$r} = $fichier; }
                        }
                    }
                }
            }
        }
    }
}
```

```

if ($#ARGV > 1) {
    $exp = $ARGV[2];
    if ($exp =~ /\^*(.*)/) {
        $exp = ".$$1";
    }
    while ($exp =~ /(.*[^\.]*)\*(.*)/ ) {
        $exp = "$1.*$2";
        $exp =~ /(.*[^\.]*)\*(.*)/;
    }
    parcours($ARGV[0], $ARGV[1], $exp);

    foreach $clef (sort keys %resultats) {
        print "$clef appariements :\n";
        foreach $clef2 (keys %{$resultats{$clef}}) { # %{ } pour indiquer qu'entre les accolades
            # il y a un tableau associatif
            print "    ", $clef2, " : ", $resultats{$clef}{$clef2}, "\n";
        }
    }
}
}

```

## Écriture d'un script CGI (9 points)

Écrivez un programme CGI en Perl qui permette d'effectuer, **pas à pas**, des calculs arithmétiques (avec seulement comme opérateurs l'addition la soustraction, la multiplication et la division).

Exemples :

