

Perl TD2

Résumé

Un descripteur (ou *handle*) de fichier est le nom d'une connexion d'E/S (Entrée/Sortie) établie entre un programme Perl actif et le monde extérieur.

Perl fournit 3 descripteurs de fichiers qui s'ouvrent chacun automatiquement sur une E/S précise :

- STDIN : entrée standard,
- STDOUT : sortie standard,
- STDERR : sortie d'erreurs standard.

Pour ouvrir tout autre descripteur de fichier, utiliser la commande `open()` :

```
open (FICDESC, "nom_fichier"); # ouvrir en lecture
open (FICDESC, ">nom_fichier") # ouvrir en ecriture
open (FICDESC, ">>nom_fichier") # ouvrir en ajout
```

Pour fermer un descripteur de fichier :

```
close (FICDESC);
```

L'utilisation est très semblable à `<>`.

Ex :

```
open (EP, "/etc/passwd");
open (RES, ">res.txt");
while (<EP>)
    chomp; # supprime le newline à la fin de $_
    print RES "I saw $_ in passwd file.";
close (EP);
close(RES);
```

L'opérateur `die()` prend une liste optionnelle de paramètres, l'affiche (à la façon de **print**) sur `STDERR` et termine le programme Perl.

Ex :

```
open (DATA, "rapport.embl") || die "Le fichier n'existe pas.";
OU
open (DATA, "rapport.embl") || die "ouverture impossible: $!";
```

(La variable `$!` contient la chaîne d'erreur décrivant l'erreur système la plus récente.)

Expressions régulières.

les expressions régulières permettent de manipuler un texte de façon très puissante. On les utilise en particulier pour tester si une chaîne de caractères comporte un certain motif.

Pour trouver une sous-chaîne :

`m/$exprreg/` : opérateur de recherche (match). Le 'm' est facultatif.

La recherche s'applique à \$_ par défaut sinon il faut utiliser l'opérateur =~

`/ $exprreg/i` : avec l'option i, recherche non sensible à la casse.

Le '.' remplace n'importe quel caractère, le '^' désigne le début de chaîne, le '\$' la fin de chaîne.

Dans une expression régulière, on peut utiliser les raccourcis suivants :

- \d : un chiffre (équivalent à [0-9])
- \D : tout sauf un chiffre (équivalent à [^0-9])
- \w : un caractère alphanumérique (équivalent à [0-9a-zA-Z_])
- \W : tout sauf un caractère alphanumérique (équivalent à [^0-9a-zA-Z_\])
- \s : un espace, équivalent à [\n\t\r\f]
- \S : tout sauf un espace, équivalent à [^\n\t\r\f]

Perl affecte 3 variables spéciales : \$&, \$' et \$', qui contiennent respectivement la partie de la chaîne qui a "matché", celle qui est avant et celle qui est après. Ainsi la position "matchante" est `length($') + 1`.

Pour récupérer certaines parties de la chaîne qui matchent des sous-expressions du motif, on place ces sous-expressions entre parenthèses. Les différents résultats sont stockés dans des variables \$1, \$2, ...

Ex :

```
if ($time = /(\d\d):(\d\d):(\d\d)/) { # heure au format hh:mm:ss
    $hours = $1;
    $minutes = $2;
    $seconds = $3;
}
```

1 Exercice 1

Considérez un fichier **source_file** contenant un texte quelconque. Ecrivez un script Perl qui recopie ce fichier dans un fichier **result_file** de sorte que chaque ligne soit précédée de son numéro de ligne.

2 Exercice 2

Le fichier **kinases_map.txt** a été récupéré depuis la banque OMIM, il contient un tableau d'informations sur les gènes de protéines kinases de l'homme. Chaque ligne contient les informations sur un gène, les différents champs sont séparés par une barre verticale (|). La description des champs est :

- Gene symbol(s),
- Gene name,
- date of entry to the database,
- Cytogenetic location (chromosome number followed by cytogenetic band(s)),

— Accession number in OMIM.

1. Ecrivez un script Perl qui lit ce fichier et affiche, séparés par des tabulations, le "gene symbol", le "gene name" et le "cytogenetic location".
2. Modifiez votre script pour qu'il affiche seulement les "tyrosine kinase genes" (i.e. ceux dont le "gene name" contient le mot tyrosine).
3. Modifiez votre script pour afficher seulement les gènes localisés sur le chromosome 9.
4. Modifiez votre script pour afficher seulement les gènes entrés dans la base de données après 1995.
5. Modifiez votre script pour afficher seulement les gènes vérifiant tous les critères précédents simultanément.