Pretty printer en `perl`

Louis Granboulan

24 janvier 2001

1 Motivations

Pour formater de jolie façon un texte, il existe des commandes unix comme `fmt` et des éditeurs comme `emacs` ou `vim` peuvent être configurés pour faire ceci automatiquement.

Il s'agit de s'arranger pour que les lignes ne soient pas trop longues (une limite de 75 ou 78 caractères est couramment utilisée, en prévision des terminaux de 80 colonnes), de vérifier l'espacement autour de la ponctuation, etc.

Le but de ce court TP est d'écrire un tel programme en `perl`. Le choix de ce langage plutôt que `C` est naturel puisqu'on devra intensivement manipuler des chaînes de caractères.

Le langage perl étant très touffu, son manuel en ligne a été découpé en plusieurs morceaux, dont voici ceux dont on se sert habituellement (tapez `man perl` pour en avoir la liste complète) : `perldata`, `perlsyn` et `perlop` pour les bases du langage, `perlre` pour la syntaxe des expressions régulières, `perlfunc` pour les fonctions prédéfinies.

2 Exercices

2.1 Imitation de `cat`

Ce petit programme perl fonctionne comme la commande `cat` : s'il est appelé sans arguments, il recopie son entrée standard sur sa sortie standard. S'il a des arguments, il considère que ce sont des noms de fichiers et, l'un après l'autre, il les lit et les recopie sur sa sortie standard.

2.2 Formatage de la ponctuation

En français nous utiliserons les conventions typographiques suivantes : lorsque les caractères ont une largeur fixe (ce qui est le cas dans un terminal) on représente toute espace par un unique caractère espace. Toute virgule est suivie d'un espace et colle au mot précédent. De même pour les points. En revanche, les points d'exclamation, d'interrogation, points-virgules et deux-points sont précédés et suivis d'un espace. La parenthèse ouvrante est précédée d'un espace et colle au mot suivant. La parenthèse fermante l'inverse. Il n'y a jamais plusieurs espaces à la suite, sauf en début de ligne.

Modifier le programme précédent pour que sa sortie standard respecte ces règles.

2.3 Ponctuation anglaise ou française

Les règles typographiques sont différentes pour la langue anglaise. Le point est suivi de deux espaces, les ponctuations ne sont jamais précédées d'un espace, sauf la parenthèse ouvrante.

Modifier le programme précédent pour qu'une option `-e` ou `-f` lui fasse utiliser respectivement les règles anglaises ou françaises.

Si vous en avez le courage, faites détecter au programme quel est le langage si ce n'est pas précisé en ligne de commande (on utilisera par exemple les dictionnaires `/usr/dict/words` et `/usr/local/util/lib/words.french`).

2.4 Formatage des paragraphes : découpage des lignes

On considère que tout ce qui est entre deux lignes vides est un unique paragraphe.

Notre programme devra désormais regrouper toutes les lignes d'un paragraphe en une seule, puis les découper entre les mots de telle sorte qu'aucune ne dépasse 75 caractères. Ce chiffre peut être paramétrable sur la ligne de commande.

On peut aussi tenir compte du fait qu'en français tous les paragraphes ont leur première ligne indentée (de deux espaces par exemple). On peut aussi vouloir appliquer à l'ensemble du paragraphe l'indentation de sa première ligne. Tout ceci devrait être paramétrable sur la ligne de commande.

2.5 Justification totale

Enfin, on veut rajouter des espaces entre les mots de telle sorte que le dernier caractère du dernier mot de chaque ligne soit en 75ème colonne, sauf pour la dernière ligne d'un paragraphe.

Si vous en avez le courage, utilisez un algorithme de rajout d'espaces qui tient compte de la structure de la langue : on rajoute volontiers des espaces après la ponctuation, on évite d'en rajouter entre un article et le nom correspondant.