Univ rsité Paris Did rot — Analys d Donné s Structuré s
L2 Informatiqu t Math-Info — Anné 2014-2015

TP n°1

Utiliser des Expressions Régulières

1 C'est quoi, `sed` ?

`sed` st l'édit ur d flux (str am **ed**itor) d Linux, qui p rm t d pars r t transform r du t xt av c un langag simpl t concis. Pour mi ux connaîtr son usag , vous vous référ r z, bi n nt ndu, à son manu l. Voici, qu lqu s x mpl s, pour comm nc r.

Un d s formats d'un app l d `sed` st l suivant:

```
sed [-options] "[ dr1[ dr2]] [comm nde ...]" [fichier ...] [> fichier_de_sortie]
```

Si vous app l z `sed` sur un fichi r, sans options, adr ss s ni command s (`sed "" toto.txt`), il lit *ligne par ligne* tout s l s lign s du fichi r, il n' n fait ri n, t il affich la lign ou la copi dans l fichi r d sorti , si vous av z précisé c lui-ci. Si vous utilis z l'option `-n`, vous arriv z à mpêch r mêm c tt écritur . Att ntion, s ul m nt l s élém nts ntr croch ts (`[...]`) sont optionn ls, notamm nt dans c cas d'abs nc d command , il faut gard r l s guill m ts (`""`) pour évit r l'ambiguïté ntr la command t l nom du fichi r.

Un façon pour r str indr la porté d'un command , lorsqu vous daign z n m ttr un , st d'utilis r l s adr ss s. Quand vous utilis z un s ul adr ss , `sed` n'opèr qu sur la (ou l s) lign (s) adr ssé (s). Vous pouv z adr ss r un lign par son numéro, ou, plus intér ssant, par l fait qu'il conti nt un motif xprimé sous form d'*expression régulière* (*basique* par défaut, t *étendue* si vous utilis z l'option `-E`) ntr d ux slashs : `/motif/`. Si vous utilis z d ux adr ss s, séparé s par un virgul , `sed` opèr sur l s lign s ntr l s d ux.

Exercice 1 La command `p` (pour **p**rint) affich la lign lu . Combin z-l av c c qu vous v n z d lir pour simul r l comport m nt d `grep`.

Télécharg z la sourc `html` d la pag w b du cours. A l'aid d doubl addr ssag t v ntu ll m nt la command `!`, affich r, séparém nt, un fois l corps (*body*) d la pag t un fois tout l r st .

Plus util qu `p`, il y a la command `s` pour **s**ubstitution. Vous pouv z l'utilis r av c l format `s/motif/rempl cement/fl gs`. L programm `sed` r mplac l s occurr nc s du `motif`, donné sous form d'un xpr ssion régulièr , par `rempl cement`. Vous pouv z modifi r l comport m nt par défaut d `sed` par l s `fl gs`.

L' ntré st lu lign par lign , t la command `s` substitu par défaut s ul m nt la pr mièr occurr nc du `motif` dans chaqu lign . Par rapport aux xpr ssions régulièr s, `sed` st dit êtr 'gourmand', c' st-à-dir , il pr nd la plus longu chaîn d caractèr s qui corr spond à l' xpr ssion.

Exercice 2 Comm nc z par fair `sed` répét r chaqu lign qu vous ntr z au clavi r, n ffaçant l pr mi r mot d chaqu lign . Pour qu `sed` lis son ntré du clavi r, il suffit d laiss r vid l champ `fichier`.

Utilis z l s *références vers l'arrière* pour répét r l' ntré n inv rsant l pr mi r t l d uxièm mot. Dans l s xpr ssions régulièr s *basiques*, vous pouv z utilis r par x mpl la command

`s/un motif\(motif 1\) utre motif\(motif 2\).../\1\2/`, t l s référ nc s v rs l'arrièr `\1` t `\2` tc. auront pour val ur l s xpr ssions qui ont *matché* r sp ctiv m nt l 1er t l 2nd motifs par nthésés, tc.

Utilis z nsuit l flag **g** (pour **g**lobal) pour r mplac r tous l s spac s par d s tir ts bas (_) t nr gistr z l résultat dans un autr fichi r (n ajoutant `> nouve u_fichier` à la fin d la lign d l'app l d `sed`).

Un d rnièr command avant d vous m ttr n rout pour attaqu r l s x rcic s plus séri ux, la command `N`. Rapp l z-vous qu `sed` trait son ntré lign par lign : `N` vous p rm t d lir un lign supplém ntair t l'ajout r à la fin d la lign déjà lu , séparé s, bi n nt ndu, par un caractèr d fin d lign (`\n`). Ainsi, vous pouv z ff ctu r d s opérations sur plusi urs lign s.

Et final m nt, sach z qu vous pouv z nchaîn r plusi urs command s d `sed` dans un s ul app l, comm vous all z voir dans l' x rcic suivant:

Exercice 3 Qu fait la command suivant?

```
sed -E "/SKIP3LINES/ {N; N; N; s/SKIP3LINES.*/Les 3 proch ines lignes ignorées./;}"
```

Exercice 4 Quand vous programm z n Java, vous êt s, bi n sûr, très pointill ux sur l s ind ntations t compagni . Notamm nt, s'il y a un s ul command après un `if`, vous n'utilis z pas d'accolad , mais vous saut z la lign , vous augm nt z l'ind ntation, t vous écriv z la command . Par x mpl :

```
if (nod != NULL)
  return (nod .g tPar nt());
```

Mais final m nt, vous chang z d'avis t décid r d'écrir c g nr d conditionn ll dans un s ul lign :

```
if (nod != NULL) return (nod .g tPar nt());
```

Utilis z `sed` pour appliqu r c tt modification au fichi r `.j v` sur l qu l vous travaill z.

2 Nettoyage de textes

Prof d français, vos élèv s vous ont r ndu 35 poèm s sous form d fichi rs t xt s. Par xpéri nc , vous sav z à l'avanc qu'il vaut mi ux qu vous comm nci z par un p tit ménag d s rr urs classiqu s afin d vous épargn r qu lqu s migrain s.

Pour fair c la d'un coup av c un s ul app l d `sed`, vous pouv z cré r un fichi r où vous m tt z tout s l s command s, t nsuit app l r `sed` sous l format suivant:

```
sed -f fichier_de_comm ndes fichier_de_poeme ...
```

Exercice 5 En vous s rvant d `sed`, automatis r l s opérations suivant s :

1. Supprim r tous l s spac s t tabulations n début d lign .
2. R mplac r l s doubl s (ou tripl s tc...) spac s au mili u du t xt par d s spac s simpl s.
3. Supprim r l s répétitions d mots (par x. *il rentra dans sa sa maison*).
4. Vérifi r qu tout s l s lign s comm nc nt bi n par un majuscul (écritur n v rs), corrig r au b soin.

5. Vérifi r qu'il n'y a jamais d' spac avant l s virgul s (mais un spac après), t qu'il y n a avant/après ";","!","?" t ":", n corrig ant c qui doit l'êtr .

Vous t st r z vos command s sur l fichi r *eleve*.

3 Une nou elle télé !

C la faisait maint nant un an qu l son d votr vi ill télé était cassé, t qu vous vous éti z habitué à r gard r tout vos films av c d s sous-titr s pour mal nt ndants. Pour Noël, vos par nts ont rach té un nouv ll télé, il va donc vous falloir adapt r vos fichi rs d sous-titr s !

Exercice 6 Vous avi z prévu d r gard r *Bon Baisers de Bruges* c soir, il vous faut donc supprim r la parti mal nt ndant du fichi rs. C ll -ci conti nt :

- d s bruitag s, sous la form `(SOUND EXPLANATION)`
- l nom d s p rsonnag s dans c rtains dialogu s, sous la form : `PERSONNAGE:` + t xt (toujours n début d lign t n majuscul s)

À l'aid d `sed`, supprim z c s élém nts du fichi r sous-titr s *In.Bruges.srt*, n sachant qu'il n'y a pas d'autr s par nthès s qu c ll s d s bruitag s.

Exercice 7 Votr film comm nc , l pop-corn st prêt, mais ô dés spoir, vous r nd z compt qu votr télé flambant n uv n support pas l s balis s html `<i>...</i>` t l s affich xplicit m nt : ri n d plus insupportabl ! Un rapid coup d'o il à la docum ntation vous inform n r vanch qu c modèl support l Markdown, t qu'un t xt n italiqu doit êtr ncadré par d ux étoil s : `*du texte*`. Problèm , afin d'épargn r votr p tit frèr d la vulgarité du film, c rtains mots sont masqués av c c mêm caractèr (ntr autr s `f***`), c qui risqu d m ttr du t xt n italiqu n'import où. À l'aid d `sed`, r mplac z l s `*` anti-vulgarités par un autr caractèr (par x mpl `@`) puis ff ctu z la substitution d s balis s italiqu s par d s étoil s (t vit , tout la famill att nd dans l canapé !)

Exercice 8 L film fini, la r st d la famill étant parti s couch r, vous vous propos z d comm nc r un nouv ll saison d votr séri favorit . Vous av z l'habitud d la r gard r av c l s sous-titr s n français, c p ndant votr nouv ll télé n support pas non plus l'UTF8... À l'aid d la command `sed y`, r mplac z tous l s caractèr s acc ntués par l ur v rsion sans acc nts dans l fichi r *The.Wire.s02e01.srt*.

4 De la musique en pagaille

Après avoir écouté n boucl un playlist sur Int rn t, vous vous décid z à la télécharg r, t voul z l'ajout r à c ll s qu vous av z déjà sur votr disqu dur. Problèm , l s formats d fichi rs n sont pas l s mêm s :
-la playlist récupéré sur int rn t st au format XSPF, dont vous pouv z obs rv r la structur dans l fichi r *playlist.xml* -vos playlists sont au format CSV, av c un lign par morc au sous la form : numéro d lign ,Artist ,Titr .

Exercice 9 En vous s rvant d `sed`, cré r un fichi r au format CSV à partir du fichi r *playlist.xml*. Vous v ill r z à c qu l s noms d s artist s ainsi qu l s titr s d s chansons ai nt un majuscul à chaqu mot.

Exercice 10 Sauriez-vous ajouter le nom du fichier mp3 correspondant à chaque ligne de votre fichier CSV ?

Exercice 11 Sauriez-vous refaire la même chose à partir du fichier *playlist2.xml* ?

5 À la chasse aux nombres premiers

On se propose dans cette dernière exercice de détecter les nombres premiers à l'aide de `sed`.

Exercice 12 Écrire dans votre langage préféré un script prenant en argument un entier n et produisant un fichier *nombres* qui contiendra 1 sur la première ligne, 11 sur la seconde, 111 sur la troisième, etc.

Exercice 13 Trouver une expression régulière correspondant aux lignes contenant un nombre (écrit en unaire) admettant un diviseur. En déduire une expression régulière pour les lignes contenant un nombre non-premier.

Exercice 14 À l'aide de la commande `sed =`, numéroter les lignes du fichier *nombres*, en séparant le numéro de ligne du contenu par une tabulation. Ainsi, la cinquième ligne du résultat obtenu devrait être :

```
5	11111
```

Exercice 15 En vous utilisant les questions précédentes, écrire un script, qui depuis le fichier *nombres*:

- numérote les lignes
- ne garde que les lignes contenant des nombres premiers
- sur chaque ligne, supprime la partie unaire.
- regroupe tous les lignes en une seule, en les séparant par des virgules.

Vous devriez ainsi obtenir la liste des nombres premiers sous la forme `2,3,5,7,11,13,17,19`.