Prise de notes avec CherryTree.

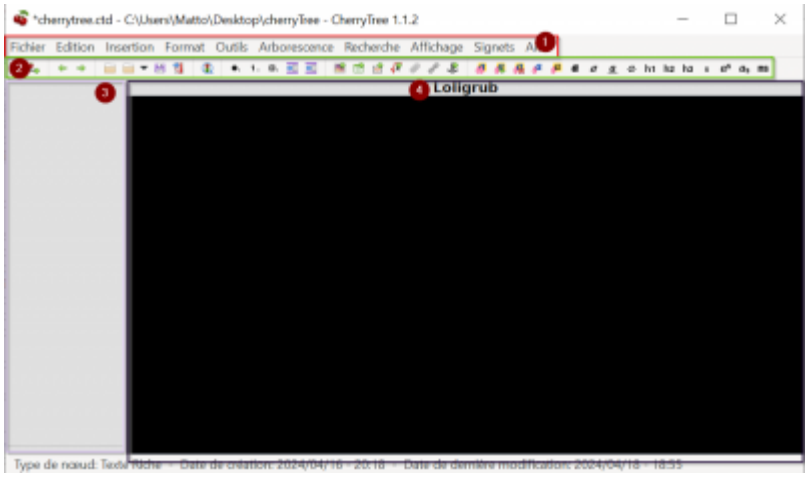
Thomas Dutat - LoLiGrUB (20 avril 2024)

Préambule

CherryTree est une application de prise de notes hiérarchique, proposant du texte riche avec une coloration syntaxique et stockant les données soit dans un seul fichier (xml ou sqlite), soit dans plusieurs fichiers et répertoires. (Sous licence GNU)

Interface

1. Barre de menus
2. Barre d'outils
3. Vue de l'arborescence
4. Zone d'édition



Vue de l'arborescence

Son arborescence est constitué de noeuds ainsi que des sous-noeuds.

Options pratique :

1. verrouiller un noeud en lecture seule (cadenas)
    - pratique pour la création de template
2. duplication de noeuds
    - copie le noeud à l'identique
3. noeud du jour
    - crée une arborescence pour la date du jour

Zone d'édition

Prise en charge des objets suivants :

- Image
- Tableau
- CodeBox
    - boîte de code exécutable
- Fichier
- Lien
- Ancre
- TOC (Table des matières)
- Horodatage
- Caractère spécial
- La règle horizontale

Enregistrement

2 type d'enregistrement

1. SQLite
    - SQLite est une base de données autonome et possède une extension .ctb ou .ctx lorsqu'elle est utilisée avec Cherrytree.
    - plus adapter aux documents volumineux
2. XML
    - XML est un langage de balisage et possède une extension .ctd ou .ctz lorsqu'il est utilisé avec Cherrytree.

Peuvent être compresser et vérouiller par un mot de passe via 7zip.

Exemple 2

Détecter un “trou” dans une séquence de fichiers:

```
ls PW1_{4700..4750}.jpg > /dev/null
```

Les redirections

Chaque programme peut rediriger la sortie écran (stdout) vers un fichier via l'opérateur >

- /dev/null : pseudo-fichier qui ignore tout ce qu'on lui envoie
- les erreurs restent visibles à l'écran: seuls sont affichés les fichiers manquants

Exemple 3

On passe la deuxième:

Dans un log de serveur web (nginx), affichage des 100 urls les plus consultées:

```
awk '{print $7}' access.log | sort | uniq -c | sort -rn | head -100
```

awk

Du nom de ses concepteurs: Al Aho, Peter Weinberger et Brian Kernighan. Outil de traitement et d'extraction de texte possédant son propre langage.

The AWK Programming Language

En gros, les mots sont séparés par un caractère blanc et sont numérotés de $1 à $x.

La ligne {print $7} signifie:

affiche le septième champ (l'url).

Le pipe

Concept clé du shell UNIX, le pipe permet de rediriger la sortie d'un programme vers l'entrée d'un autre.

Il est représenté par le caractère |

Exemple 4: encore awk

Dans un log de serveur web (nginx), affichage des 30 urls générant le plus de code http 404:

```
awk '$9 == "404" {print $7}' access.log | sort | uniq -c | sort -rn | head -n 30
```

La ligne $9 == “404” {print $7} signifie:

Si le neuvième champ de chaque ligne est 404, alors affiche le septième champ.

Exemple 5

Générateur de phrase de passe composées de 2 mots:

```
look . | grep -E "^[a-z]{4,8}$" | shuf | head -40 | xargs -n2
```

look

Outil (apparu dans l'édition 7 de UNIX) permettant de rechercher un mot dans un fichier.

Si aucun fichier n'est spécifié, recherche dans un dictionnaire.

. signifie “n'importe quel terme”

grep

Get Regular Expression and Print: recherche toute chaîne répondant à l'expression régulière et

l'affiche.

[quote, Wikipedia] Chaîne de caractères, qui décrit selon une syntaxe précise, un ensemble de chaînes de caractères possibles.

N'importe quel mot de 4 à 8 lettres

- ^ : rien avant
- [a-z] : n'importe quelle lettre de a jusqu'à z
- {4,8} : répétée de 4 à 8 fois
- $: rien après

xargs

Parfois, il n'est pas possible que la sortie d'une commande corresponde à l'entrée d'une autre. xargs permet de se sortir de situations parfois difficiles où il n'est pas possible d'enchainer les commandes avec des |

Par défaut, xargs affiche ce qu'il reçoit sur 1 ligne. L'argument -n2 lui indique de grouper 2 éléments par ligne.

```
cd /usr/bin ; ls -1 | shuf | xargs man
```

Autre exemple d'expression régulière

Afficher toutes les lignes qui ne sont pas des commentaires dans un fichier de configuration:

```
grep '^[^#]' /etc/pacman.conf
```

- ^ : rien avant
- [^x] : qui n'est pas le caractère x.

Du fun, du fun, du fun

Ça ne sert à rien, mais c'est tellement bien !

Les outils indispensables:

- Mettez de la couleur dans vos terminaux: lolcat
- Inspectez votre machine: neofetch
- Réalisez des bannières avec style: figlet
- Invitez une vache dans le terminal: cowsay
- Faites parler chuck norris: fortune-mod-chucknorris

La météo

Rapide, facile et sans pub:

```
curl fr.wttr.in/Boussu
```

curl: outil d'interrogation de serveur web en ligne de commande.

Base combo

```
neofetch
catimg loligrub-asbl.png
chuck | cowsay | lolcat -F 1
figlet -tc -f shadow "Merci de votre attention \!" | lolcat
```

La sortie de figlet peut être redirigée vers /etc/motd (message of the day).

Ultra combo !

```
yes "$(seq 231 -1 16)" | while read i; do printf "\x1b[48;5;${i}m\n";\
sleep .03; done
```

```
grep -ao "[/\\]" /dev/urandom | sed -e 's,\\,\,' -e 's,/,╱,' |\
tr -d \\n | lolcat -F 0.001
```

Le meilleur pour la fin

```
for p in {36..1..4}; do espeak-ng -v en -p $p\
"We are the Borg. Lower your shields and surrender your ships...\
Your biological and technological distinctiveness will be added to our own.\
Resistance is futile."\
& sleep 0.007; done
```

```
yes $COLUMNS $LINES|awk 'BEGIN{x=y=e=f=1}{if(x==$1||!x)\
{e*=-1};if(y==$2||!y){f*=-1};x+=e;y+=f;\
printf "\033[%s;%sH",y,x;system("sleep .02")}'
```

Quelques références

- The AWK Programming Language
- Regex Cheat Sheet
- Bash Guide
- Shell mon amour
- Command-line Tools can be 235x Faster than your Hadoop Cluster

Merci

Questions ?