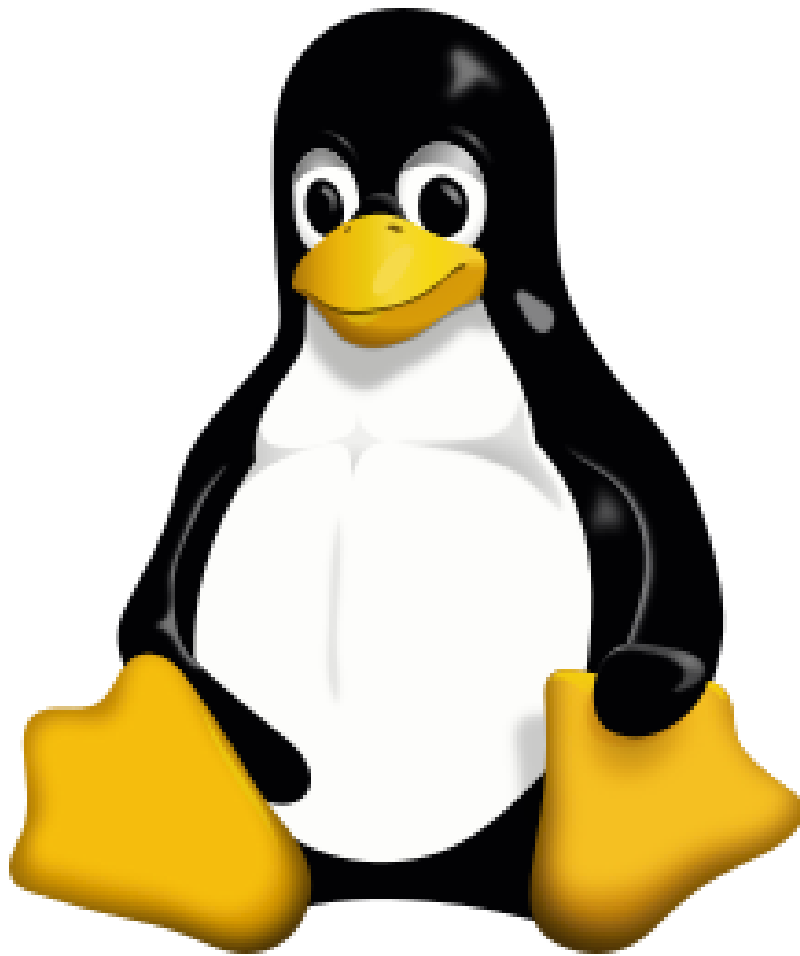


Rapport : Création d'un moteur de blog

AUBIER Clément, MOREIRA RIBEIRO Nuno



Projet Système R1.04

21/11/2022

Table des matières

❖ Un besoin, un logiciel	2
❖ Manuel d'utilisation	3
➤ Prise en main	3
➤ Éditer un fichier markdown	4
➤ Supprimer un fichier édité	4
➤ Lister les fichiers édités	5
➤ Convertir en HTML/PDF	5
➤ Visualiser les fichiers convertis	6
❖ Organisation et difficultés rencontrées	8
➤ Répartition des tâches	8
➤ Difficultés surmontées et expérience	9
➤ Améliorations possibles et limites	10

Un besoin, un logiciel

Il nous a été demandé de développer un logiciel permettant de créer des sites statiques. Il faut donc pouvoir éditer du texte, le sauvegarder puis le convertir en page html, afin qu'il soit utilisable sur le web. De plus, on doit pouvoir visualiser le rendu de la page sur la machine. Nous avons choisi le langage bash pour créer ce logiciel car ce qui est demandé n'est pas complexe, et peut se coder facilement avec les outils mis à disposition par ce langage shell.

Le script est ainsi assez concis. Il permet d'abord d'éditer un fichier au format markdown, grâce à l'éditeur de notre choix installé sur la machine. Une fois la rédaction terminée, il enregistre ce fichier dans un dossier markdown, avec les autres textes écrits grâce au logiciel. On peut également supprimer et visualiser ces fichiers directement depuis l'application.

De plus, le logiciel propose de convertir nos fichiers écrits en markdown en fichiers html ou PDF. Les pages html pourront donc être intégrées individuellement dans un site web, ou un blog. Tandis que le fichier PDF rassemble l'entièreté des pages éditées à l'aide de ce logiciel. Enfin, l'application peut nous permettre de visualiser les pages html et le PDF créés.

Manuel d'utilisation

Voici de courtes instructions qui vous permettront de prendre rapidement en main le logiciel

Prise en main

Après avoir téléchargé le logiciel, qui se présente sous la forme d'un fichier exécutable **blogue.sh**, ouvrez un terminal à son emplacement. Assurez-vous d'avoir les droits en exécution pour ce fichier en tapant **chmod +x blogue.sh**, puis exécutez-le via la commande **./blogue.sh**. Cela va afficher la liste des commandes disponibles du logiciel.

```
nmoreiraribe@info-211-14x:~/projetSysteme/ProjetSystem $ ./blogue.sh
[Erreur] 1 action est attendue
Usage : ./blogue.sh COMMANDE [PARAMETRE]

La liste des commandes est editer, supprimer, lister, construire

- Éditer un document : ./blog.sh editer [PAGE]
- Supprimer un document : ./blog.sh supprimer [PAGE]
- Lister les documents : ./blog.sh lister
- Générer les fichiers PDF/HTML : ./blog.sh construire
- Visualiser le site : ./blog.sh visualiser [pdf]
```

Éditer un fichier markdown

Lors de la première exécution, un fichier *markdown* sera automatiquement créé dans le même répertoire que *blogue.sh*. Il contient tous les fichiers éditables créés avec le logiciel.

Tapez **`./blogue.sh editer [nom du fichier]`** pour créer votre premier fichier. Attention: l'extension du fichier étant implicite, il ne faut pas la marquer. Si vous ne précisez aucun nom de fichier, alors le logiciel vous affichera la liste des textes déjà édités et vous invitera à en choisir un.

L'éditeur qui sera utilisé par défaut est celui contenu dans la variable d'environnement **\$EDITOR** de Linux. Si cette variable n'est pas définie, alors c'est l'éditeur contenu dans **/usr/bin/editor** qui est utilisé. Enfin, si aucun éditeur ne s'y trouve, plusieurs éditeurs standards sont recherchés sur le système afin de pouvoir ouvrir le fichier.

```
nmoreiraribe@info-211-14x:~/projetSysteme/ProjetSystem $ ./blogue.sh editer
markdown/fic1.md markdown/fic2.md markdown/fic3.md markdown/index.md
Voici le(s) fichier(s) que vous pouvez éditer.
Saisissez un nom de page à éditer : █
```

Supprimer un fichier édité

Vous pouvez supprimer un fichier édité avec l'application en tapant **`./blogue.sh supprimer [nom du fichier]`**. Si aucun nom de fichier n'est donné, l'application affichera la liste des fichiers supprimables. Dans tous les cas, une confirmation de suppression vous sera demandée.

```
nmoreiraribe@info-211-14x:~/projetSysteme/ProjetSystem $ ./blogue.sh supprimer
markdown/fic1.md markdown/fic2.md markdown/fic3.md markdown/index.md
Voici le(s) fichier(s) que vous pouvez supprimer.
█
```

Lister les fichiers édités

Pour afficher la liste des fichiers markdown édités à l'aide de l'application, entrez **./blogue.sh lister**. Cette fonctionnalité ne demande aucun paramètre supplémentaire.

```
nmoreiraribe@info-211-14x:~/projetSysteme/ProjetSystem $ ./blogue.sh lister
markdown/fic1.md markdown/fic2.md markdown/fic3.md markdown/index.md
```

Convertir en HTML/PDF

L'application peut se charger elle-même de la conversion des fichiers markdown en HTML et PDF. Pour cela, tapez **./blogue.sh construire** ; aucun argument supplémentaire n'est demandé. L'ensemble des fichiers édités avec l'application seront reconstruits à chaque fois que vous exécuterez la commande.

HTML

Un dossier *web* est généré automatiquement à l'emplacement de *blogue.sh*. Il contient vos fichiers markdown convertis au format HTML. Pour faciliter la navigation entre les pages créées, un fichier *index* est généré dans le dossier markdown, puis converti dans la dossier *web*. Il contient les liens vers les autres pages HTML créées ainsi que des informations sur la date et l'utilisateur qui l'a généré.

```
nmoreiraribe@info-211-14x:~/projetSysteme/ProjetSystem $ ./blogue.sh construire
nmoreiraribe@info-211-14x:~/projetSysteme/ProjetSystem $ ls
blog.pdf  blogue.sh  markdown  README.md  web
nmoreiraribe@info-211-14x:~/projetSysteme/ProjetSystem $ cd web/
nmoreiraribe@info-211-14x:~/projetSysteme/ProjetSystem/web $ ls
fic1.html fic2.html fic3.html index.html
nmoreiraribe@info-211-14x:~/projetSysteme/ProjetSystem/web $
```

PDF

Un fichier *blog.pdf* est créé au même emplacement que l'application. Il regroupe l'ensemble des pages éditées en un seul fichier PDF de plusieurs pages. C'est très utile pour garder un historique des pages éditées.

```
nmoreiraribe@info-211-14x:~/projetSysteme/ProjetSystem $ ls  
blog.pdf  blogue.sh  markdown  README.md  web
```

Visualiser les fichiers convertis

Il est possible d'ouvrir les fichiers construits avec la commande **./blogue.sh visualiser [pdf]**. Si aucun paramètre n'est fourni, alors l'application ouvre le fichier *index.html*, qui contient le lien des pages HTML générées, à l'aide du navigateur par défaut de l'utilisateur.

```
nmoreiraribe@info-211-14x:~/projetSysteme/ProjetSystem $ ./blogue.sh visualiser  
nmoreiraribe@info-211-14x:~/projetSysteme/ProjetSystem $
```

[fic1](#)

[fic2](#)

[fic3](#)

Contenu actualisé le sam. 03 déc. 2022 21:41:48 CET par clement sur ubuntu15dk1

Si l'argument "pdf" est fourni, alors c'est le fichier *blogue.pdf* qui est ouvert, avec la visionneuse de PDF par défaut de l'ordinateur.

```
nmoreiraribe@info-211-14x:~/projetSysteme/ProjetSystem $ ./blogue.sh visualiser pdf
```

ceci est le contenu du fic 1

voici le contenu du fic 2

ci-après le contenu du fic 3

Organisation et difficultés rencontrées

Répartition des tâches

Création d'un moteur de Blogue

Cadre : Projet de d'intro Système - IUT Informatique
Groupe : AUBIER Clément, MOREIRA Nuno

Date de début: Mar. 15/11/2022

Tâches	Attribué à	Semaine 1	Semaine 2	Semaine 3
Prise de connaissance du besoin		L M M J V S D L	M M J V S D L	M M J V S D L
Explication du sujet	M. GIOT			
Découpage du travail	Clément, Nuno			
Code des 1 ^{ers} fonctions (main, usage, choice)	Clément, Nuno			
Programmation des fonctions				
Editor, choseEditor → édition de fichier	Nuno			
list → lister les fichiers	Nuno			
delete, arkForDelete → suppression de fichiers	Clément			
Apprendre à se servir de pandoc & xdg-open	Clément, Nuno			
build, buildPdf, buildHtml → construire	Clément			
Ajustements et rédaction du rapport				
Mise en commun	Clément, Nuno			
Correction de bogues	Clément, Nuno			
Fonction findEditor	Clément, Nuno			
Rédaction du rapport	Clément, Nuno			
Mise en page du rapport	Clément, Nuno			
Remise sur Moodle	Clément, Nuno			

Difficultés surmontées et expérience

Au cours de la phase de développement, nous avons fait face à de nombreux problèmes. Le premier fut le suivant : comment faire passer les arguments donnés lors de l'exécution du programme dans chaque sous-fonction ? Après réflexion, on a considéré que le mieux à faire était d'appeler chaque fonction avec l'argument "\$@" , se passant ainsi tous les arguments les uns les autres.

On a également dû résoudre le problème d'accès à une variable inexistante. En effet, on ne peut pas lire le contenu de "\$1" si aucun argument n'est fourni au script. Il a donc fallu tester, à chaque fois qu'on avait besoin d'un argument, que celui-ci était bien fourni à l'aide de "\$#". Enfin, nous avons pris un certain temps avant de comprendre comment fonctionne l'outil pandoc, permettant de convertir les fichiers.

Néanmoins, ce projet nous a énormément apporté. Il nous a notamment appris à répartir équitablement les tâches et à utiliser l'outil Github pour mettre en commun le code. De plus, ça nous a également permis de mettre à profit nos connaissances théoriques concernant le langage bash. Il a fallu réfléchir au découpage des fonctions, à leur structure globale puis les intégrer à l'application.

Améliorations possibles et limites

Le temps imparti pour créer le logiciel fut assez court et il a donc été assez difficile de rajouter des fonctionnalités supplémentaires au logiciel. Cependant, si on avait eu le temps, on aurait bien aimé effectuer des tests pour s'assurer que l'utilisateur ne fournit jamais d'extensions au moment de manipuler les noms de fichiers. En effet, l'application aurait pu le faire elle-même.

De plus, on aurait aimé pouvoir personnaliser le contenu des documents avec des variables d'environnement et différents thèmes disponibles. Il aurait également été possible de créer un fichier *index.html* plus stylisé, avec une structure complète et des éléments CSS, au lieu de simples liens.