



Rush du module d'UNIX

Pipex

42 staff staff@42.fr

Résumé: Ce projet est la découverte dans le détail et par la programmation d'un mécanisme d'UNIX que vous connaissez déjà.

Table des matières

I	Préambule	2
II	Sujet	3
II.1	Les objectifs	3
II.2	Détails	4

Chapitre I

Préambule

Comme il s'agit d'un rush, vous ne trouverez aucune indication dans l'e-learning. Vos amis sont toujours les mêmes (à savoir vos voisins de gauche, de droite, ou inversement, les man, google, etc...). A vous de trouver sur ces 2 jours comment réaliser ce qui est demandé. Attention, il y a plusieurs niveaux de complexité. De nombreux phénomènes se produisent lorsque certaines erreurs surviennent. Vous devrez les gérer conformément aux réactions du système.

Chapitre II

Sujet

II.1 Les objectifs

Votre objectif est de réaliser le programme pipex. Ce programme fonctionne de la façon suivante :

```
$> ./pipex file1 cmd1 cmd2 file2
```

file1 et file2 sont des noms de fichiers.

cmd1 et cmd2 sont des commandes shell avec leurs paramètres.

L'exécution du programme pipex a le même effet que la commande shell suivante :

```
< file1 cmd1 | cmd2 > file2
```

Votre objectif est de réaliser le même mécanisme qui est mis en place par le shell (on veut pas voir d'étape intermédiaire ou de découpage, tout doit marcher d'un seul coup).

Exemples :

```
(clark@almarita) ./pipex infile `ls -l` `wc -l` outfile
```

executera la même chose que "< infile ls -l | wc -l > outfile" en sh

```
(clark@almarita) ./pipex infile `grep a1` `wc -w` outfile
```

executera la même chose que "< infile grep a1 | wc -w > outfile" en sh

II.2 Détails

Rendu classique sur le git, Makefile classique, et bien sûr, seul le contenu de votre dépôt sera utilisé en soutenance. Un fichier auteur est à rendre à la racine de votre dépôt. Il sera rempli selon les consignes habituelles.

Seules sont autorisées : malloc, free, et tous les appels systèmes de la libc. Attention, pas les fonctions de la libc.