

Compte rendu TP SHELL

Nos réalisations comprennent :

Différents types de commandes tels que les commandes internes :

- *cd* : Le changement de répertoire (*cd*, *cd..*, *cd ~*, *cd /*).
- *fg* : La mise en premier plan de commandes (*fg 1*, *fg cmd*).
- *bg* : La mise en arrière plan de commandes (*bg 1*, *bg cmd*)..
- *jobs* : L'affichage des commandes en arrière plan.
- *quit/exit* : L'arrêt du shell.

les commandes externes comme :

- *wc*
- *grep*
- *ls*
- *kill*
- ...

La communication entre les commandes :

- *|* : La redirection d'entrées/sorties de commandes avec les pipes (tubes).
- *<* : La redirection d'un fichier en entrée.
- *>* : La redirection de sortie vers un fichier.

La gestion des signaux :

- *Ctrl+c* : Provoque la mort du processus actif au sein du shell.
- *Ctrl+z* : Provoque l'arrêt du processus actif au sein du shell.

Autres fonctionnalités :

- *&* : La mise en arrière plan de commandes.
- *~/home* : L'affichage de l'environnement courant.

Vous pourrez aussi trouver dans notre dossier de rendu, la documentation complète du code, créée avec doxygen.

Pour chaque fonctionnalité implémentée nous avons fait plusieurs tests :

- Entrées valides : qui nous permettent de vérifier le bon déroulement des commandes ainsi que la validité de leurs sorties.
- Entrées invalides : qui nous permettent de vérifier la bonne gestion des erreurs.

Fonctionnement de nos pipes :

Dès le début du projet nous avons décidé de gérer les pipes avec seulement deux identifiants de pipes : *prevCmdPipe* et *nextCmdPipe*.

- *prevCmdPipe* : contient les descripteurs de fichiers d'entrée et de sortie venant de la commande précédente.
- *nextCmdPipe* : contient les descripteurs de fichiers d'entrée et de sortie allant vers la commande suivante.

Le tout est contenu dans un tableau nommé *pipes*.

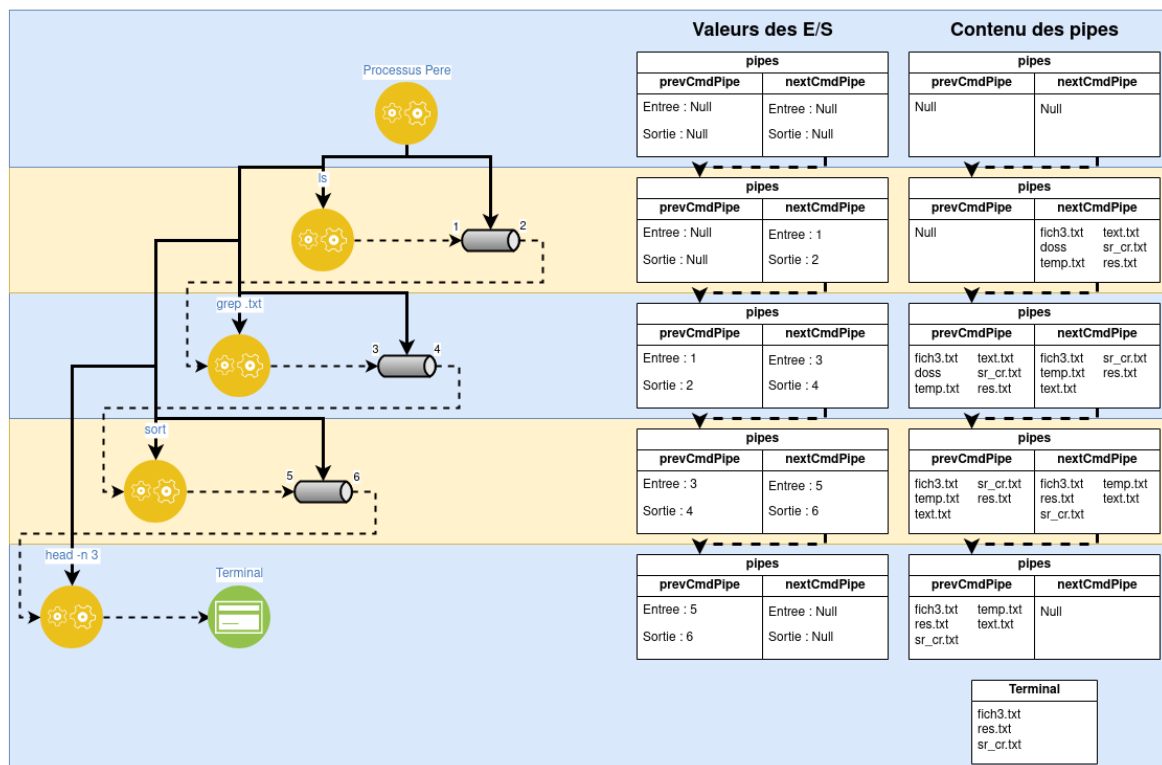
Dans le diagramme ci dessous :

- La valeur des entrées/sorties correspond au numéro d'un descripteur de fichier.
- Le contenu des pipes correspond à ce que l'on peut lire dans la sortie du pipe.
- L'entrée d'un processus et d'un pipe se situe sur la gauche et leur sortie sur la droite.
- On exécute dans un dossier contenant les fichiers/répertoires suivants :
 - *fich3.txt*
 - *doss*
 - *temp.txt*
 - *text.txt*
 - *sr_cr.txt*
 - *res.txt*

la commande suivante :

- `ls | grep .txt | sort | head -n 3`

Diagramme de la création et de l'utilisation des pipes



Le fait d'avoir seulement deux identifiants de pipe nous contraint à faire des échanges de valeurs entre eux.

Chaque processus lit dans la sortie de *prevCmdPipe* et écrit dans l'entrée de *nextCmdPipe*. Après la création du processus, dans le père, on assigne à *prevCmdPipe*, *nextCmdPipe*. Puis on crée un nouveau pipe que l'on assigne à *nextCmdPipe*.

Les commandes en bout de ligne (***ls*** | *grep .txt* | *sort* | ***head -n 3***) agissent différemment, celle en début de ligne (***ls***) vérifie si un fichier redirigé en entrée. Si c'est le cas alors le processus lit dans le fichier donné, sinon, il lit dans l'entrée standard. La sortie est envoyée vers la commande suivante (*grep .txt*) dans le pipe *nextCmdPipe*.

De même, la commande en fin de ligne (***head -n 3***) vérifie si la sortie est redirigée vers un fichier. Si c'est le cas alors le processus écrit dans le fichier donné, sinon, il écrit dans l'entrée standard.