

## MINISHELL

### Objectif :

Le but de ce nouveau projet est de **créer un simple shell** (tout comme une petit bash or zsh), permet d'appréhender les processus et les descripteurs de fichiers.

### Goal :

*The goal of this new project is to create a simple shell (just like a small bash or zsh), which will help you apprehended processes and file descriptors.*

---

### Histoire :

L'existence du shell est directement liée à l'existence de la technologie de l'information. Du fait des difficultés de communication avec un ordinateur, c'est-à-dire utilisation des commutateurs 0/1 alignés, les informaticiens ont très vite compris qu'il était important de trouver une manière plus simple de communiquer avec ces derniers. C'est ainsi qu'ils ont développé des langages quelque peu proches de l'anglais composé de lignes de commandes interactives.

### History :

*The existence of shells is directly linked to the existence of IT. Due to the difficult communication with computers, based on aligned 0/1 switches, computer scientists quickly understood the importance of finding a "simple" way to communicate with them. Therefore, they decided to create languages, somewhat close to English, using interactive lines of commands.*

**I. Instructions Communes / Common Instructions :**

- Le projet **doit respecter la Norme**, y compris la partie bonus / *The project must be written in accordance with the Norme, including bonuses ;*
- **Aucunes fonctions ne doit s'interrompre de façon inattendue** / *Your functions should not quit unexpectedly ;*
- **Toutes mémoires allouées doit être libérées si nécessaire** / *All heap allocated memory space must be properly freed when necessary ;*
- Si le sujet le demande, le **Makefile doit compiler les sources dans l'output demandé avec les flags -Wall, -Werror et -Wextra sans relink** (le Makefile en question doit au moins contenir les règles \$(NAME), all, clean, fclean et re) / *If the subject requires it, Makefile must compile all source files to the required output with the flags -Wall, -Wextra and -Werror; without any relink (the Makefile must at least contain the rules \$(NAME), all, clean, fclean and re) ;*
- Si le sujet permet l'utilisation d'une **libft** personnelle, ses sources et son Makefile doit être rangé dans un fichier libft et doit être **associé au Makefile du projet en question** / *if the subject allows you to use a personal libft, it's sources and Makefile must be put into a folder libft and must be associated to the Makefile of the current project.*

## II. Partie Obligatoire / Mandatory Part :

### 1. Aperçu rapide / Quick overview :

Nom du programme / <i>Program name</i>	<b>Minishell</b>
Description	Ecrire un Shell / <i>Write a shell</i>
Makefile	Oui / <i>Yes</i>
Libft	Oui / <i>Yes</i>
Fonctions externes autorisées / <i>Externals function authorized</i>	<ul style="list-style-type: none"> <li>- readline ;</li> <li>- rl_clear_history / rl_on_new_line / rl_replace_line / rl_redisplay ;</li> <li>- add_history ;</li> <li>- printf / write ;</li> <li>- malloc / free ;</li> <li>- open / read / close ;</li> <li>- fork ;</li> <li>- wait / waitpid / wait3 / wait4 ;</li> <li>- signal / kill ;</li> <li>- exit ;</li> <li>- getcwd ;</li> <li>- chdir ;</li> <li>- stat / lstat / fstat ;</li> <li>- unlink ;</li> <li>- execve ;</li> <li>- dup / dup2 ;</li> <li>- pipe ;</li> <li>- opendir / readdir / closedir ;</li> <li>- strerror / perror ;</li> <li>- isatty ;</li> <li>- ttyname / ttyslot ;</li> <li>- ioctl ;</li> <li>- getenv ;</li> <li>- tcsetattr / tcgetattr / tgetent / tgetflag / tgetnum / tgetstr / tgoto / tputs.</li> </ul>

2. En details / In details :

Ton Shell devrait / *Your shell should :*

- **Pas interpréter des guillemets non fermés ni des caractères spéciaux non spécifiés comme “ \ ” ou “ ; ”** / *Not interpret unclosed quotes or unspecified special characters like “ \ ” or “ ; ”.*
- **Pas utiliser plus d'une variable globale** / *Not use more than one global variable.*
- **Montrer un prompt en attendant une commande** / *Show a prompt when waiting for a new command.*
- **Posséder un historique fonctionnel** / *Have a working history.*
- **Rechercher et lancer le bon exécutable** (basé sur la variable PATH ou en utilisant un chemin relatif ou absolu) / *Search and launch the right executable (based on the PATH variable or by using relative or absolute path).*
- Avoir les **builtins** suivant / *Must implement the following builtins :*
  - **Echo** : avec l'option -n / *with the option -n ;*
  - **Cd** : seulement avec un chemin relatif ou absolu / *with only a relative or absolute path ;*
  - **Pwd** : sans option / *with no option ;*
  - **Export** : sans option / *with no option ;*
  - **Unset** : sans option / *with no option ;*
  - **Env** : sans option ou arguments / *with no option or arguments ;*
  - **Exit** : sans option / *with no option.*
- **“ ‘ ” doit inhiber toute interprétation d'une séquence de caractères** / *“ ‘ ” must inhibit all interpretation of a sequence of characters.*
- **“ “ “ doit inhiber toute interprétation d'une séquence de caractères sauf “ \$ ”** / *“ “ “ must inhibit all interpretation of a sequence of characters except for \$.*
- Gérer les **redirections** suivantes / *Must manage the following redirections :*
  - **< “** : redirectionne les entrées / *redirect inputs ;*
  - **> “** : redirectionne les sorties / *redirect outputs ;*
  - **<< “** : lire les entrées à partir de la source actuelle jusqu'à ce qu'une ligne contenant uniquement le délimiteur apparaisse (pas besoin de mettre à jour l'historique) / *read input from the current source until a line containing only the delimiter is seen (doesn't need to update history) ;*
  - **>> “** : redirectionne les sorties avec le mode d'ajout / *redirect output with append mode.*

- En ce qui concerne les **pipes**, “|”, **la sortie de chaque commande dans le pipeline est connectée via un tuyau au saisie de la commande suivante** / *With regard to the pipes, “|”, the output of each command in the pipeline is connected via a pipe to the input of the next command.*
- Les **variables d’environnement** (“\$” suivi de caractères) **devraient s’étendre à leur valeur** / *Environment variables (“\$” followed by characters) should expand to their value.*
- “\$?” **devrait développer jusqu’à l’état de sortie du premier plan exécuté le plus récemment pipeline** / *“\$?” should expand to the exit status of the most recently executed foreground pipeline.*
- Ctrl-C, ctrl-D et ctrl-\ doivent fonctionner comme en bash / *Ctrl-C, ctrl-D and ctrl-/ should work like in bash :*
  - **Ctrl-C** : afficher un nouveau prompt sur la nouvelle ligne / *print a new prompt on a newline ;*
  - **Ctrl-D** : quitter le shell / *quit the shell ;*
  - Ctrl-\ : rien / *nothing.*
- **La fonction readline peut produire des fuites de mémoires mais ces dernières ne doivent pas être gérées** / *Readline function can produce some leak you don't need to fix this.*

### III. Builtins :

#### 1. Echo :

Description	<b>Écrit des arguments dans la sortie standard</b> / <i>Write arguments to the standard output.</i>
Spécification / <i>Specification</i>	Echo écrit tous les opérandes spécifiés, <b>séparées par des un espace (" ") et suivant d'un retour à la ligne ("\n")</b> , dans la sortie standard / <i>The echo utility writes any specified operands, separated by single blank ( ' ') characters and followed by a newline ('\n') character, to the standard output.</i>
Synopsis	<b>echo [-n] [char *, ...] =&gt; echo [-n][char **]</b>
Option	L'option <b>-n</b> écrire des arguments dans la sortie standard sans le retour à la ligne / <i>Option -n will write arguments in the standard output without the final newline.</i>

```

→ Test ls -la
total 32
drwxr-xr-x  4 mboy  2020   136 Sep 10 14:44 .
drwxr-xr-x  3 mboy  2020   102 Sep 10 14:42 ..
-rwxr-xr-x  1 mboy  2020  8432 Sep 10 14:44 a.out
-rw-r--r--  1 mboy  2020   112 Sep 10 14:45 test_echo.c
→ Test echo a.out test_echo.c
a.out test_echo.c
→ Test echo -n a.out test_echo.c
a.out test_echo.c
→ Test

```

2. Cd:

Description	<b>Change le répertoire de travail / Change the working directory</b>
Spécification / <i>Specification</i>	<p>Cd va se charger de changer le répertoire de travail de l'environnement d'exécution du shell actuel / <i>Cd will change the working directory of the current shell execution environment</i> :</p> <ul style="list-style-type: none"> <li>- <b>Step 1</b> : Si aucun opérande de répertoire est donné et que la variable d'environnement HOME est vide ou indéfinie, rien ne se produit / <i>if no operand is given and the HOME environment variable is empty or undefined, nothing will happen.</i></li> <li>- <b>Step 2</b> : Si aucun opérande de répertoire est donné mais que la variable d'environnement HOME est définie sur une variable non vide, cd va prendre cette dernière comme opérande de répertoire / <i>If no directory operand is given but the HOME environment variable is set to a non-empty variable, cd will take this as the directory operand.</i></li> <li>- <b>Step 3</b> : Si l'opérande de répertoire commence par un slash (" / "), on attribuera au chemin actuelle, connu sous le nom de curpath, l'opérande en question et passera à l'étape 7 / <i>If the directory operand begins with a slash (" / "), the current path, named curpath, will be set to the given operand and shall move on to step 7.</i></li> <li>- <b>Step 4</b> : Si le premier composant de l'opérande de répertoire n'est ni un point ou un double points, on procède à l'étape 6 / <i>If the first component of the directory operand is neither a dot or a dot-dot, proceed to step 6.</i></li> <li>- <b>Step 5</b> : ?</li> <li>- <b>Step 6</b> : Attribuer au chemin actuel, curpath, l'opérande de répertoire donnée / <i>Set the current path, curpath, the given directory operand.</i></li> <li>- <b>Step 7</b> : Le chemin actuel, curpath, va être ensuite convertie sous forme canonique, afin d'analyser chaque composant du début à la fin dans le sens donné / <i>The current path, curpath, will then be converted into its canonical form in order to consider each component from beginning to end, in sequence</i> : <ul style="list-style-type: none"> <li>→ Les composants "point" et tout slash délimitant le composant sont retirés / <i>all dot and slash components are deleted</i> ;</li> <li>→ Pour tout composant double-point, si est seulement si le composant précédent est ni la racine ni une double points, cd retournera un message d'erreur si le composant précédent n'est pas un répertoire, ou, dans le cas</li> </ul> </li> </ul>

	<p>contraire, supprimera tous les caractères slash séparant le composant précédent au double points, et tous les caractères slash séparant les double points du composant suivant (le cas échéant) / <i>For all dot-dot composants, if and only if the preceding composant in nor the root nor a dot-dot, cd will return a error message if the composant is not a directory, or, otherwise it will delete all slash characters separating the preceding composant from the dot-dot, and all slash characters separating the dot-dot from the following component will also be deleted.</i></p> <p>→ Pour simplifier la démarche, l'on peut également : supprimer tous les caractères slash de fin qui ne sont pas également des caractères slash de début, remplacer plusieurs caractères slash consécutifs par un seul caractère slash, remplacer au moins 3 caractères slash de début par un seul (si après cette étape de canonisation, le chemin actuel est nul, le processus s'arrête la) / <i>To simplify the task, you can also : delete all slash characters that are not leading slash characters, replace all multiple non-leading consecutive slash characters by a single one, replace three or more leading slash character by a single one (if, as a result of this canonicalization, the current path in null, th process stops here).</i></p> <ul style="list-style-type: none"> <li>- <b>Step 7</b> : Si le chemin actuel est plus grand que {PATH_MAX} octets (en incluant le null final) et si l'opérande de répertoire n'était plus plus grand que {PATH_MAX} octets (en incluant de nouveau le null final), alors le chemin actuel sera converti d'un nom de chemin absolue en un nom de chemin relatif équivalent si possible (cette conversion doit toujours être considérée si la valeur de PWD est une sous-chaîne initiale du chemin actuel, soit curpath, les implémentations peuvent également appliquer cette conversion si le chemin actuel n'est pas plus grand que {PATH_MAX} octets ou si l'opérande de répertoire était plus long que {PATH_MAX} octets) / <i>If the current path is longer than {PATH_MAX} bytes (including the terminal null) and the directory operand was not longer than {PATH_MAX} bytes (including the terminal null), then the curpath will be converted from an absolute pathname to an equivalent relative pathname if possible (this conversion should always be considered if the value of PWD is une substring og curpath, implementations may also apply this conversion if the curpath is not longer than {PATH_MAX} bytes or if the directory operand was longer than {PATH_MAX} bytes).</i></li> </ul>
--	--



	<p>- <b>Step 8</b> : Finalement, cd va alors effectuer des actions équivalentes à la fonction chdir() appelée avec le curpath comme argument de chemin, si une de ces actions échoués pour une raison ou une autre, cd renvoie un message d'erreur approprié et le processus s'arrêtera là / <i>Finally, cd will perform actions equivalent to the chdir() function called with curpath as the path argument, if an error appears during any of those actions, cd will return an appropriate error message and will stop the process.</i></p> <p>Si, durant l'une de ces étapes, la variable PWD est redéfinie, la variable d'environnement OLDPWD sera remplacé par le précédent PWD / <i>If, during one of those steps the PWD variable is redefined, the OLDPWD environment variable shall be replaced by the prior PWD.</i></p>
Synopsis	<b>cd [directory] =&gt; cd [char *]</b>
Définition / Definition	<p>Chemin / <i>Path</i> : un chemin d'accès est une liste de noms de répertoire séparés par des barres obliques suivis d'un nom de répertoire ou d'un nom de fichier / <i>a path is a liste of directory names separated by slashes, followed by the name of a directory of a file.</i></p> <p>Chemin Absolue ou Complet / <i>Absolute Path</i> : Un chemin absolu commence par une lettre de lecteur suivie par double point (ex : D:.) / <i>An absolute path starts with a drive letter followed by a colon (ex : D:.).</i></p> <p>Chemin Relatif / <i>Relative Path</i> : Un chemin relatif fait référence à un emplacement qui est relatif à un répertoire courant, c'est derniers utilisent des simples points et des doubles points qui correspondent au répertoire courant et au répertoire parent (deux points sont utilisés pour monter d'un niveau dans la hiérarchie, un point unique représente le répertoire courant lui-même / <i>A relative path refers to a location that is relative to the current directory, these uses dots and dot-dot to refers to the current directory and the parent directory (dot-dot are used to go up a level in the hierarchy, a single one represents the current directory itself.</i></p>
<pre> → Desktop cd → ~ cd Desktop/42/SoLong → SoLong git:(master) ✗ cd .. → 42 cd ../MyLibft/Sources/GNL → GNL cd ../../../../../../Downloads → Downloads cd → ~ █ </pre>	

3. Pwd :

Description	<b>Retourne le nom du répertoire de travail en cours</b> / <i>Returns the name of the current working directory.</i>
Spécification / <i>Specification</i>	Pwd affiche le nom complet du répertoire en cours, <b>c'est-à-dire que tous les composants du chemin affiche seront de vrais répertoires</b> , aucun lien symbolique ne sera présent (chemin absolue) / <i>Pwd will return the complete name of the working directory, this is to say that all components of the shown path are real directories, no symbolic links will be present (absolute path).</i>
Synopsis	<b>pwd</b>
<pre> → ~ pwd /Users/mboy → ~ cd Desktop/42/Minishell/Builtin → Builtin pwd /Users/mboy/Desktop/42/Minishell/Builtin → Builtin █ </pre>	

4. Export :

Description	<b>Définie l'attribut d'exportation pour les variables</b> / <i>Set the export attribute for variables</i>
Spécification / Specification	<b>Export va afficher toutes les variables de l'environnement</b> / <i>Export will display all the variables of the environment</i>
Synopsis	<b>export / export name[=word] =&gt; export [char *[= char *]]</b>
Option	<b>L'option name[=word] signifie que le shell doit donner à l'attribut export aux variables correspondant au noms spécifiés</b> , ce qui les amènera à être dans l'environnement des commandes exécutées par la suite, si name est suivi par =word, alors la valeur de cette variable prendra la valeur de word / <i>The option name[=word] means that the shell will give the export attribute to the variables corresponding to the specified names, which shall cause them to be in the environment of the subsequently executed commands, if the name of the variable is followed by =name, then the value of the variable will be set to name.</i>

```

➔ ~ export ARG=hello
➔ ~ export
ARG=hello
Apple_PubSub_Socket_Render=/private/tmp/com.apple.launchd.KyTqVxeT8W/Render
COLORFGBG='7;0'
COLORTERM=truecolor
COMMAND_MODE=unix2003
HOME=/Users/mboy
ITERM_PROFILE=Default
ITERM_SESSION_ID=w0t0p0:C68776FD-7164-4EF0-A850-1E106256770B
LANG=en_US.UTF-8
LC_TERMINAL=iTerm2
LC_TERMINAL_VERSION=3.4.8
LESS=-R
LOGNAME=mboy
LSCOLORS=Gxfxcxdxbxegedabagacad
OLDPWD=/Users/mboy
PAGER=less

```

5. Unset :

Description	Unset va “unset” les valeurs et attributs des variables et fonctions / <i>Unset will unset values and attributes of variables and functions</i>
Spécification / Specification	Unset va, en quelque sorte, annuler ce que export a créer / <i>Export will, in a way, undo what export has create</i>
Synopsis	<b>unset [name] =&gt; unset [char *]</b>

```

➔ ~ export ARG=HELLO
➔ ~ unset ARG
➔ ~ export
Apple_PubSub_Socket_Render=/private/tmp/com.apple.launchd.KyTqVxeT8W/Render
COLORFGBG='7;0'
COLORTERM=truecolor
COMMAND_MODE=unix2003
HOME=/Users/mboy
ITERM_PROFILE=Default
ITERM_SESSION_ID=w0t0p0:C68776FD-7164-4EF0-A850-1E106256770B
LANG=en_US.UTF-8
LC_TERMINAL=iTerm2
LC_TERMINAL_VERSION=3.4.8
LESS=-R
LOGNAME=mboy
LSCOLORS=Gxfxcxdxbxegedabagacad
OLDPWD=/Users/mboy
PAGER=less
PATH=/Users/mboy/.brew/bin:/Users/mboy/.brew/bin:/Users/mboy/.brew/bin:/Users/
Users/mboy/.brew/bin:/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/n
PWD=/Users/mboy
SECURITYSESSIONID=186a8
SHELL=/bin/zsh
SHLVL=1
SSH_AUTH_SOCK=/private/tmp/com.apple.launchd.V7kp1DaZGI/Listeners
TERM=xterm-256color
TERM_PROGRAM=iTerm.app
TERM_PROGRAM_VERSION=3.4.8
TERM_SESSION_ID=w0t0p0:C68776FD-7164-4EF0-A850-1E106256770B
TMPDIR=/var/folders/zz/zyxvpxvq6csfxvn_n0000hth00046m/T/
USER=mboy
XPC_FLAGS=0x0
XPC_SERVICE_NAME=0
ZSH=/Users/mboy/.oh-my-zsh
__CF_USER_TEXT_ENCODING=0x0:0:0

```

6. Env :

Description	
Spécification / <i>Specification</i>	
Synopsis	
Option	

7. Exit :

Description	Quitte le shell / Exit the shell
Synopsis	<b>exit</b>

#### IV. Écrire son propre Shell / Writing your own Shell

Le Shell est un programme qui interagit avec l'utilisateur à travers un terminal (ou prends les input d'un fichier et exécute les commandes qui sont passées au Système Opérateur). / *Shell is a program that interacts with the user through a terminal (or takes the input from a file and executes a sequence of commands that are passed to the Operating System).*

##### 1. Programme Shell / Shell Program :

Un programme Shell est une application qui va nous permettre d'interagir avec l'ordinateur. Dans un programme Shell, l'utilisateur peut faire tourner des programmes ainsi que redirectionner un input et un output pour qu'ils viennent d'un fichier. Un programme Shell fournit également des constructions de programmation telles que if, for, while, functions, variables, etc. De plus, on peut y trouver des caractéristiques telles que de l'édition de ligne, un historique, complétion du dossier, les caractères génériques, l'expansion des variables d'environnement et les constructions de programmation / *Shell program is an application that will allow us to interact with a computer. In a Shell, the user will be able to run programs and also redirect an input and an output to come from a file. A Shell program also provides programming constructions such as if, for, while, functions, variables, etc. In addition, a Shell program offers features such as line editing, file wildcards, environment variable expansion, and programing constructions.*

<b>sh</b>	Shell Program	Programme shell original en UNIX / <i>Original Shell programme in UNIX.</i>
<b>csh</b>	C Shell	Version améliorée de sh / <i>Improved version of sh.</i>
<b>tcsh</b>	//////////	Version de csh avec l'édition de ligne / <i>Version of csh with line editing</i>
<b>ksh</b>	Korn Shell	Père de tous les plus Shell avancés / <i>Father of all advanced Shells.</i>
<b>bash</b>	GNU Shell	Prend le meilleur de tous les programmes Shells, il est actuellement le programme Shell le plus courant / <i>Takes all the best of all Shell programs, it is currently the most common Shell program.</i>

A noter, en addition des Shell de commandes de lignes, il existe également des Shell Graphique qui simplifient l'utilisation d'un ordinateur pour la plupart des utilisateurs (comme Windows, Desktop, MacOS Finder, Linux Gnome, etc). Cependant, ces Shells graphiques ne peuvent pas servir de substitution aux Shells de commandes de lignes pour certains utilisateurs qui souhaitent exécuter des séquences complexes de commandes de façon répété ou avec des paramètres non disponibles dans ces version graphiques d'un programme Shell / *Note, that in addition to command-line Shells, there is also graphical Shells that simply the use of computers*

*for most users (such as Desktop, MacOS Finder, Linux Gnome, etc.). However, these graphical Shells can't be used as substitutes to command line Shells for users who want to execute complex sequences of commands repeatedly or with parameters not available in these graphical Shells.*



2. Les 3 parties composantes d'un program Shell / All 3 components of a Shell program:

Le Parseur / <i>The Parser</i>	
<p>Le parseur est un composant logiciel qui lit des ligne comme "ls -la" et va mettre dans une structure data appelée "table de commandes" qui va stocker les commandes qui vont être exécuté / <i>The parser est un software component that reads the command of line such as "ls -la" and that will puts into a data structure called "command table" that will store the commands that will be executed.</i></p>	<p>Un parseur est divisé en deux parties : -</p> <ul style="list-style-type: none"> <li>- L'analyste lexical ou Lexer qui prend l'input caractères et les rassemblent pour former des mots appelés tokens / <i>A Lexical Analyzer or Lexer that takes the input characters and put them together into words called tokens ;</i></li> <li>- Le parser qui traite les tokens en accordances avec la grammaire afin de créer la "table de commandes" / <i>The parser that processes the tokens according to grammar and that builds the "command table".</i></li> </ul>
L'Exécuteur / <i>The Executor</i>	
<p>L'exécuteur va prendre la table de commandes générée par le parseur et pour chaque commande simple dans stockés va créer un nouveau processus. Il créera également, si nécessaire, des pipes qui vont communiquer l'output d'un processus à l'input du prochain. De plus, l'exécuteur va redirectionner l'input standard, l'output standard et l'erreur standard s'il y a des redirections / <i>The executor will take the command table generated by the parser and will create for each simple command stocked a new process. It will also create, if necessary, pipes that let communicate an output for one process to the input of the following process. In addition, the executor will redirect the standard input, the standard output and the standard error if there are any redirections.</i></p>	<p>Exemple / <i>Example</i> :</p> <p style="text-align: center;">A   B   C   D &gt; outfile &lt; infile 2 &gt; errfile</p> <ul style="list-style-type: none"> <li>- S'il y a une redirection tel que "&lt; infile" qui est détectée par le parser, l'input de la premier commande simple A sera redirectionné par infile / <i>If there is a redirection such as "&lt; infile" is detected by the parser, the input of the first simple command A is redirected from infile.</i></li> <li>- S'il y a une redirection tel que "&gt; outfile", l'output de la dernière commande simple D sera redirectionner a outfile / <i>If there is a redirection such as "&gt; outfile", the output of the last simple command D will be redirected to outfile.</i></li> <li>- S'il y a une redirection tel que "&gt;&amp; errfile" le stderr de toutes les commandes simples processus seront redirection vers errfile / <i>If there is a redirection sur as "&gt;&amp; errfile" the stderr of all simple commands processes will be redirected to errfile.</i></li> </ul>

Sous-Système Shell / <i>Shell Subsystems</i>
<p>On peut également y trouver des sous-systèmes qui vont venir compléter le Shell /  <i>You can also find subsystems that complete the Shell :</i></p> <ul style="list-style-type: none"> <li>- Variables d'environnement / <i>Environment variables</i> :  Des expressions sous la forme <math>\\${VAR}</math> sont étendu à la variable d'environnement correspondante (le Shell doit également capable de définir, d'étendre et de environnement d'impression vars) / <i>Expressions of the form <math>\\${VAR}</math> are expanded with the corresponding environment variable (the Shell should also be able to set, expand and print environment vars).</i></li> <li>- Caractères génériques / <i>Wildcards</i> :  Des arguments sont la forme de <math>a*a</math> sont étendu à tous les fichiers qui leurs correspondent dans le répertoire local et de multiples autres répertoires / <i>Arguments of the forme <math>a*a</math> are expanded to all files that match them in the local directory and multiple other directories.</i></li> <li>- Sous-Shell / <i>SubShells</i> :  Des arguments entre guillemets sont exécutés et leurs outputs sont envoyés sous forme d'input du Shell / <i>Arguments between back sticks are executed and there output is sent as input to the Shell.</i></li> </ul>