

minishell_subject_ko.pdf

Minishell

As beautiful as a shell

шел만큼이나 아름다운 그것

Summary: The objective of this project is for you to create a simple shell. Yes, your own little bash or zsh. You will learn a lot about processes and file descriptors.

요약: 이번 프로젝트의 목표는 간단한 셸을 만드는 것입니다. 맞아요, 당신만의 작은 bash나 zsh이요. 프로세스나 파일 디스크립터에 대해 아주 많이 배우게 될 겁니다.

Contents

Aa Chapter	≡ Contents	# page
I	<u>Introduction</u>	2
II	<u>Common Instructions</u>	3
III	<u>Mandatory part</u>	4
IV	<u>Bonus part</u>	6

Chapter 1

Introduction

The existence of shells is linked to the very existence of IT. At the time, all coders agreed that communicating with a computer using aligned 1/0 switches was seriously irritating . It was only logical that they came up with the idea to communicate with a computer using interactive lines of commands in a language somewhat close to english .

헬은 IT의 태초부터 함께했습니다. 그 당시, 모든 개발자들은 1과 0으로만 이루어진 스위치로 컴퓨터와 통신하는 것은 굉장히 성가시다는 점에 동의했어요. 그리고 영어와 비슷한 언어로 작성된 명령어의 줄을 이용해 컴퓨터와 소통한다는 아이디어를 떠올린 것도 당연한 수순이었죠.

With Minishell, you'll be able to travel through time and come back to problems people faced when Windows didn't exist.

Minishell과 함께라면, Windows가 존재하지 않았을 시절 사람들이 겪었던 문제를 마주해볼 수 있을 겁니다.

Chapter 2

Common Instructions

- ***Your project must be written in accordance with the Norm. If you have bonus files/functions, they are included in the norm check and you will receive a 0 if there is a norm error inside.***

프로젝트는 Norm 규칙에 맞춰 작성되어야 합니다. 보너스 파일/함수가 존재할 경우, 그 또한 norm 검사에 포함되며 norm error가 있을 시 0점을 받게 됩니다.

- ***Your functions should not quit unexpectedly (segmentation fault, bus error, double free, etc) apart from undefined behaviors. If this happens, your project will be considered non functional and will receive a 0 during the evaluation.***

함수들은 정의되지 않은 행동들과 별개로 예기치 않게 중단되어서는 안 됩니다.(예를 들어, segmentation fault, bus error, double free 등) 만약 이렇게 중단되면, 당신의 프로젝트는 작동하지 않는 것으로 여겨지고 평가에서 0점을 받을 것입니다.

- ***All heap allocated memory space must be properly freed when necessary. No leaks will be tolerated.***

필요한 경우 heap에 할당된 모든 메모리 공간은 적절하게 해제되어야 합니다. 메모리 누수는 용납될 수 없습니다.

- ***If the subject requires it, you must submit a Makefile which will compile your source files to the required output with the flags -Wall, -Wextra and -Werror, and your Makefile must not relink.***

과제에서 필요한 경우, **-Wall -Wextra -Werror** 플래그를 지정하여 컴파일을 수행하는 **Makefile**을 제출해야 합니다. Makefile은 relink 되어서는 안 됩니다.

- ***Your Makefile must at least contain the rules \$(NAME), all, clean, fclean and re.***

Makefile은 최소한 **\$(NAME), all, clean, fclean, re** 규칙을 포함해야 합니다.

- ***To turn in bonuses to your project, you must include a rule bonus to your Makefile, which will add all the various headers, librairies or functions that are forbidden on the main part of the project. Bonuses must be in a different file _bonus.{c/h}. Mandatory and bonus part evaluation is done separately.***

프로젝트에 보너스를 제출하려면, Makefile에 **보너스 규칙**을 포함해야 합니다. 이 보너스 규칙은 프로젝트의 메인 파트에서 금지되었던 모든 다양한 헤더, 라이브러리, 또는 함수들을 추가하여야 합니다. 보너스 과제는 반드시 **_bonus.{c/h}**라는 별도의 파일에 있어야 합니다. 반드시 수행하여야 하는 메인 파트의 평가와 보너스 파트의 평가는 별도로 이뤄집니다.

- ***If your project allows you to use your libft, you must copy its sources and its associated Makefile in a libft folder with its associated Makefile. Your project's Makefile must compile the library by using its Makefile, then compile the project.***

만일 프로젝트에서 여러분의 libft 사용을 허용한다면, 소스들과 관련 Makefile을 함께 루트 폴더 안에 있는 libft 폴더에 복사해야 합니다. 프로젝트의 Makefile은 우선 libft의 Makefile을 사용하여 라이브러리를 컴파일한 다음, 프로젝트를 컴파일해야 합니다.

- ***We encourage you to create test programs for your project even though this work won't have to be submitted and won't be graded. It will give you a chance to easily test your work and your peers' work. You will find those tests especially useful during your defence. Indeed, during defence, you are free to use your tests and/or the tests of the peer you are evaluating.***

이 과제물을 제출할 필요가 없고, 채점 받을 필요가 없을지라도, 저희는 여러분들이 프로젝트를 위한 테스트 프로그램을 만들 것을 권장합니다. 이것은 여러분의 과제물과 동료들의 과제물을 쉽게 테스트할 수 있게 도울 것입니다. 또한, 평가를 진행할 때 이러한 테스트 프로그램들이 특히 유용하다는 사실을 알게 될 것입니다. 평가 시에는 여러분의 테스트 프로그램과 평가 받는 동료의 테스트 프로그램들을 당연히 자유롭게 사용할 수 있습니다.

- **Submit your work to your assigned git repository. Only the work in the git repository will be graded. If Deepthought is assigned to grade your work, it will be done after your peer-evaluations. If an error happens in any section of your work during Deepthought's grading, the evaluation will stop.**

할당된 git 저장소에 과제물을 제출하세요. 오직 git 저장소에 있는 과제물만 등급이 매겨질 것입니다. Deepthought가 평가하는 과제의 경우엔, 동료평가 이후에 Deepthought가 수행됩니다. 만약 Deepthought 평가 중에 오류가 발생한다면, 그 즉시 평가는 중지될 것입니다.

Chapter 3

Mandatory part

<u>Aa</u> Title	≡ 프로 그램 이름	≡ minishell
<u>Untitled</u>	제출할 파일	
<u>Untitled</u>	Makefile	만들어야 함
<u>Untitled</u>	인자	
<u>Untitled</u>	사용가능한 외부 함수	readline, rl_on_new_line, rl_replace_line, rl_redisplay, add_history, printf, malloc, free, write, open, read, close, fork, wait, waitpid, wait3, wait4, signal, kill, exit, getcwd, chdir, stat, lstat, fstat, unlink, execve, dup, dup2, pipe, opendir, readdir, closedir, strerror, errno, isatty, ttyname, ttyslot, ioctl, getenv, tcsetattr, tcgetattr, tgetent, tgetflag, tgetnum, tgetstr, tgoto, tputs
<u>Untitled</u>	직접 만든 libft	사용 가능
<u>Untitled</u>	설명	셸을 만드세요

Your shell should:

당신의 셸은:

- **Not interpret unclosed quotes or unspecified special characters like \ or ;.**
닫히지 않은 따옴표나 특정되지 않은 특수문자 (\나 ; 등...) 을 해석하지 않아야 합니다.

- ***Not use more than one global variable, think about it and be ready to explain why you do it.***

전역변수는 한 개 이상을 사용할 수 없으며, 왜 전역변수를 사용했는지 깊게 생각해 보고 그 이유를 설명할 수 있어야 합니다.

- ***Show a prompt when waiting for a new command.***

새로운 명령어를 입력할 수 있는 프롬프트를 보여줘야 합니다.

- ***Have a working History.***

작업 히스토리를 갖고 있어야 합니다.

- ***Search and launch the right executable (based on the PATH variable or by using relative or absolute path)***

(PATH 변수나 상대, 절대 경로를 활용하여) 올바른 실행 파일을 찾아 실행할 수 있어야 합니다.

- ***It must implement the builtins:***

다음의 내장 기능을 실행할 수 있어야 합니다:

- ***echo with option n***
 - n 옵션을 사용할 수 있는 echo
- ***cd with only a relative or absolute path***

오직 상대 또는 절대경로만 사용하는 cd
- ***pwd with no options***

옵션이 없는 pwd
- ***export with no options***

옵션이 없는 export
- ***unset with no options***

옵션이 없는 unset
- ***env with no options or arguments***

옵션이나 인자값이 없는 env
- ***' inhibit all interpretation of a sequence of characters.***

- `'` 는 일련의 문자열에 대한 해석을 금지합니다.
- **" inhibit all interpretation of a sequence of characters except for \$.**
 - `"` 는 `$` 를 제외한 모든 문자열에 대한 해석을 금지합니다.
- **Redirections:**

리다이렉션:

 - `<` **should redirect input.**
 - `<` 는 입력을 리다이렉션 하여야 합니다
 - `>` **should redirect output.**
 - `>` 는 출력을 리다이렉션 하여야 합니다
 - `<<` **read input from the current source until a line containing only the delimiter is seen. it doesn't need to update history!**
 - `<<` 는 현재 소스에서 구분자를 포함한 줄을 만나기 전까지 입력값을 읽어들이고, 기록을 업데이트할 필요는 없습니다!
 - `>>` **should redirect output with append mode.**
 - `>>` 는 출력을 추가 모드로 리다이렉션합니다.
- **Pipes | The output of each command in the pipeline is connected via a pipe to the input of the next command.**

파이프 `|` : 각 파이프라인마다 명령어의 출력값은 파이프로 연결되어 다음 명령어의 입력값으로 들어가야 합니다.
- **Environment variables (\$ followed by characters) should expand to their values.**

환경 변수 (`$` 다음에 문자열이 오는 형식) 은 그들의 값으로 확장되어야 합니다.
- **\$? should expands to the exit status of the most recently executed foreground pipeline.**
- `$?` 는 가장 최근에 실행한 포그라운드 파이프라인의 종료 상태를 확장하여야 합니다
- `ctrl-C`, `ctrl-D` and `ctrl-\` **should have the same result as in bash.**
 - `ctrl-C`, `ctrl-D`, `ctrl-\` 는 bash와 동일하게 동작하여야 합니다.
- **When interactive:**

상호작용이 가능할 때:

- **`ctrl-C` print a new prompt on a newline.**

`ctrl-C` 는 새로운 줄에 새로운 프롬프트를 출력합니다

- **`ctrl-D` exit the shell.**

`ctrl-D` 는 셸을 종료합니다.

- **`ctrl-\` do nothing.**

`ctrl-\` 은 아무런 동작도 하지 않습니다.

Anything not asked is not required.

요구하지 않은 것들을 만드실 필요는 없습니다.

For every point, if you have any doubt take bash as a reference

과제의 어떠한 지점에서든, 의문점이 생긴다면 bash를 참고하세요.

Chapter 4

Bonus part

- ***If the Mandatory part is not perfect don't even think about bonuses***

필수 파트가 완벽하지 않으면 보너스는 생각하지도 마세요.

- **`&&`, `||` with parenthesis for priorities.**

괄호를 이용해 우선순위를 표현한 `&&`, `||`

- **`the wildcard *` should work for the current working directory.**

와일드카드 `*` 가 현재 작업 디렉토리에 대해 동작하여야 합니다.