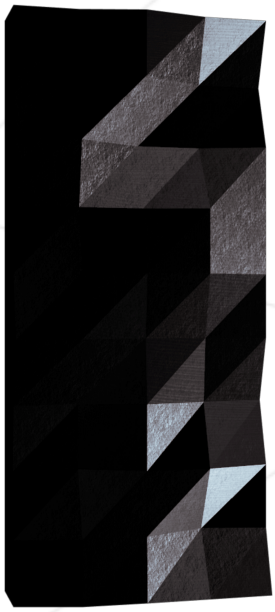

Microshell Piscine

microshell-01

Summary: This document is the subject for the microshell-01 of the Microshell Piscine @ 42 Tokyo.

Contents

Chapter I

Instructions

- Your project must be written in C.
- Only this page will serve as reference; do not trust rumors.
- Watch out! This document could potentially change up to an hour before submission.
- These exercises are carefully laid out by order of difficulty - from easiest to hardest. We `will not` take into account a successfully completed harder exercise if an easier one is not perfectly functional.
- Make sure you have the appropriate permissions on your files and directories.
- You have to follow the submission procedures for every exercise.
- Your exercises will be checked and graded by your fellow classmates.
- You cannot leave any additional file in your directory than those specified in the subject.
- Got a question? Ask your peer on the right. Otherwise, try your peer on the left.
- Your reference guide is called `Google / man / the Internet / ...`.
- Examine the examples thoroughly. They could very well call for details that are not explicitly mentioned in the subject...
- Instruction which are not written or not shown on the example are considered undefined. you should define those undefined behavior reasonably.
- Segmentation Fault or other unexpected termination of a program(double free, infinite loop) should not happen. If it occurs, your grade will be `0` during evaluation.
- No memory leak are allowed. If it occurs, your grade will be `0` during evaluation.
- 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, use gcc.
- Your Makefile must at least contain the rules $(NAME), all, clean, fclean and re. If it doesn't compile with these flags, your grade will be `0` during evaluation.

- 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.
- Your project must compile and executed on guacamole.42tokyo.jp. If It doesn't compile or execute on guacamole.42tokyo.jp, your grade will be `0` during evaluation.

Chapter II

Foreword

`commands...?`

Chapter III

Exercise 00 : absolute path command

	Exercise 00
absolute path command	
Turn-in directory : $ex00/$	
Files to turn in : `*`	
Allowed functions : `write, read, malloc, free, exit, execve, fork, waitpid, perror, stat, access`	

Create a program which meets the following requirements.

- Accept string from command line argument and use execve to execute as command.
- Implement necessary functionality so that the program behave as shown in the example below.

  Example)

```
?> ./microshell-01 /bin/hostname
c1r1s1.42tokyo.jp
?> ./microshell-01 hostname
command not found: hostname
?> ./microshell-01 '/bin/hostname -s'
c1r1s1
?> ./microshell-01 'hostname -s'
command not found: hostname
```

Chapter IV

Exercise 01 : absolute || relative path command

	Exercise 01
absolute || relative path command	
Turn-in directory : $ex01/$	
Files to turn in : `*`	
Allowed functions : `write, read, malloc, free, exit, execve, fork, wait, wait3, wait4, waitpid, perror, stat, access, getenv`	

Create a program which meets the following requirements.

- Implement previously required features.
- Implement necessary functionality so that the program behave as shown in the example below.

  Example)

```
?> env -i PATH=/bin/ ./microshell-01 /bin/hostname
c1r1s1.42tokyo.jp
?> env -i PATH=/bin/ ./microshell-01 hostname
c1r1s1.42tokyo.jp
?> env -i PATH=/bin/ ./microshell-01 '/bin/hostname -s'
c1r1s1
?> env -i PATH=/bin/ ./microshell-01 'hostname -s'
c1r1s1
?> env -i PATH=/usr/bin/:/bin/ ./microshell-01 'hostname -s'
c1r1s1
?> env -i ./microshell-01 'hostname -s'
command not found: hostname
```

Chapter V

Exercise 02 : chain commands

	Exercise 02
chain commands	
Turn-in directory : *ex*02/	
Files to turn in : *	
Allowed functions : `write, read, malloc, free, exit, execve, fork, wait, wait3, wait4, waitpid, perror, stat, access, getenv, dup2, close, pipe`	

Create a program which meets the following requirements.

- Implement previously required features.
- Implement necessary functionality so that the program behave as shown in the example below.

  Example)

```
?> env -i PATH=/bin/ ./microshell-01 /bin/hostname
c1r1s1.42tokyo.jp
?> env -i PATH=/bin/ ./microshell-01 hostname
c1r1s1.42tokyo.jp
?> env -i PATH=/bin/ ./microshell-01 '/bin/hostname -s'
c1r1s1
?> env -i PATH=/bin/ ./microshell-01 'hostname -s'
c1r1s1
?> env -i ./microshell-01 'hostname -s'
command not found: hostname
?> env -i PATH=/bin/:/usr/bin/ ./microshell-01 hostname wc
1       1      18
?> env -i PATH=/bin/:/usr/bin/ ./microshell-01 hostname -s wc
command not found: -s
?> env -i PATH=/bin/:/usr/bin/ ./microshell-01 'hostname -s' wc
1       1       7
```

Chapter VI

Exercise 03 : chain commands with timeout

	Exercise 03
chain commands with timeout	
Turn-in directory : $ex03/$	
Files to turn in : `*`	
Allowed functions : `write, read, malloc, free, exit, execve, fork, wait, wait3, wait4, waitpid, perror, stat, access, getenv, dup2, close, pipe, signal, sigaction, alarm`	

Create a program which meets the following requirements.

- Implement previously required features.
- Implement necessary functionality so that the program behave as shown in the example below.

  Example)

```
?> env -i PATH=/bin/ ./microshell-01 'sleep 3' 'echo done'
done
?> env -i PATH=/bin/ ./microshell-01 1 'sleep 3' 'echo done'
Timeout!
?> env -i PATH=/bin/ ./microshell-01 -1 'ls -lR' wc
command not found: -1
```

Chapter VII

Bonus

	Exercise 04
more builtin	
Turn-in directory : $ex04/$	
Files to turn in : *	
Allowed functions : *	

Create a program which meets the following requirements.

- Implement previously required features.
- Implement other `options` which improve users experience. (For example: verbose mode, handle whitespace character)
- For each `option` which improve user's experience, it will be graded 1point.(MAX 5points)

  Example)

```
?> env -i PATH=/bin/:/usr/bin/ ./microshell-01 -v 'hostname -s' wc
[1/2] hostname -s
c1r1s1
[2/2] wc
1       1       7
?> env -i PATH=/bin/:/usr/bin/ ./microshell-01 --verbose 'hostname -s' wc
[1/2] hostname -s
c1r1s1
[2/2] wc
1       1       7
```