Programming Assignment #2

Due: October 30th (Tue), 11:59 PM

1. Introduction

The goal of this assignment is for you to familiarize yourself with process creation and signal handling. In this assignment, we make a program that collects the title of Internet websites.

2. Problem specification

Implement the `collect()` function in the `collect.c` file. The prototype of the `collect()` function is as follows.

```
void collect (void);
```

Make a program that receives and processes a web page address and a command from standard input (`stdin`). The main function of the program is to download the input web page using the `wget` command and to collect the title of the web site from the web page file.

There are two types of strings to be processed: a web page address or a command.

The web page address is processed as specified in Section 2.1 and commands are processed as specified in Section 2.2. Otherwise, it is regarded as an invalid input string and is processed as specified in Section 2.3.

2.1 Web pages

A web page address is a string beginning with "`http://`" or "`https://`".
The tasks to be performed are as follows.

- Create a child process that downloads the page using the `wget` command.
  - ✓ Save the output of the `wget` command to a file. The name of the file to be downloaded can be set as an option of the `wget` command. Specify the name corresponding to 〈sequence number〉.
  - ✓ For example, if you are currently processing the second web page address of a command, the name of the file is to be saved "2".
- Open the file you downloaded with the `wget` command and find the title. If there is more than one title, collect the first appearing title. There are no cases where the title does not exist.
- Print the collected title string to standard output according to the following format.

```
printf("%d>%s:%s\n", sequence, domain, title);
```

- ✓ `sequence`: Indicates the current processing sequence. Starting from 1, it increments by 1 each time a web page address or a command is processed.
- ✓ `domain`: If the rightmost part (TLD) of the domain address is not ".kr", from right to second string is the domain, and if the rightmost part of the domain address is ".kr", from right to third string is the domain. (Note: **Section 3.3**)
- ✓ `title`: Found title string

- Input/Output Example 1

```
http://www.skku.edu/eng                          /* standard input */
1>skku.edu:Sungkyunkwan University (SKKU)        /* standard output */

http://www.skku.edu                              /* standard input */
2>skku.edu:성균관대학교                           /* standard output */

http://icc.skku.ac.kr                            /* standard input */
3>skku.ac.kr:성균관대학교 정보통신대학              /* standard output */
```

2.2 Commands

There are four types of commands.

- `print`: After receiving a domain as an argument, it prints the longest title among those of the corresponding domains collected so far.
- `stat`: Print the number of titles collected so far.
- `load`: After receiving a file name as an argument, it reads and processes the web page addresses or commands written in the file.
- `quit`: Exit the program.

2.2.1 `print`

The `print` command prints the longest title among the titles of the corresponding domain. Based on the above **Example 1**, the longest title for the `skku.edu` domain is `Sungkyunkwan University (SKKU)`. The criterion for 'the longest title' is the number of bytes required to represent a string. Consider that Hangul characters require more than one byte to represent a single character.

- If there are titles of the same length, print the first collected title.
  - ✓ An address that can be received as an argument is always a domain, not a sub-domain.
  - ✓ The output format is the same as the web pages.
  - ✓ If there is no collected title for the domain received as an argument, print **"Not Available"**.

- The `print` command should always be done synchronously. The `print` command must be executed after all web page addresses or commands received previously have all completed.
- Input/Output Example 2

```
print skku.edu                                  /* standard input */
4>skku.edu:Sungkyunkwan University (SKKU)       /* standard output */

print google.com                                /* standard input */
5>Not Available                                 /* standard output */
```

2.2.2 `stat`

Prints the number of titles successfully collected so far without errors. Prints the number of duplicate titles collected for the same web page address.

- The `stat` command should always be executed synchronously. This must be done after the web page address or command received before the `stat` command has been completed.
- The output format is as follows.

```
printf("%d>%d titles\n", sequence, count);
```

- Input/Output Example 3

```
stat                                            /* standard input */
6>3 titles                                      /* standard output */
```

2.2.3 `load`

It processes web page addresses or commands written in the file until the file received as an argument returns EOF.

- Each time a web page address or command is processed, the 〈sequence number〉 is incremented.
- Example of a file to be received as an argument (e.g., `command.txt`)

```
http://skku.edu
print skku.edu
stat
```

- Input/Output Example 4

```
load command.txt                                /* standard input */
7>skku.edu:성균관대학교                          /* standard output */
8>skku.edu:Sungkyunkwan University (SKKU)       /* standard output */
9>4 titles                                      /* standard output */
```

2.2.4 `quit`

Exit the program. If a child process exists, it forcibly terminates all child processes via the `SIGKILL` signal.

2.3 Errors

- When the input string is neither a web page address nor a command.
  - ✓ Ignore the input string and keep the 〈sequence number〉.
- When an error occurs while executing the `wget` command.
  - ✓ Increase the 〈sequence number〉 and print "`Error occurred!`".
- Input/Output Example 5

```
http://ThisSiteDoesNotReallyExist.com/haha    /* standard input */
10>Error occurred!                            /* standard output */
```

2.4 Child process creation

- When you download a file using the `wget` command, you can choose the following two ways.
  - ✓ Always create a single child process and execute the `wget` command.
  - ✓ Create multiple child processes and execute the `wget` command concurrently.
- Input/Output Example 6
  - ✓ Suppose you have three inputs and created three child processes to execute the `wget` command.
  - ✓ If the titles are collected in the order of the second(2nd), third(3rd), and first(1st) web page, print the titles in the order they were collected.
  - ✓ However, the 〈sequence number〉 must follow the input order.

```
http://www.skku.edu/eng                       /* standard input */
http://www.skku.edu                           /* standard input */
http://icc.skku.ac.kr                         /* standard input */
2>skku.edu:성균관대학교                        /* standard output */
3>skku.ac.kr:성균관대학교 정보통신대학          /* standard output */
1>skku.edu:Sungkyunkwan University (SKKU)     /* standard output */
```

- The top 5 programs that execute normally with the fastest execution time will earn 20% bonus points.

2.5 Signals

- This program does not terminate upon the receipt of a `SIGINT` signal.

- When the `SIGUSR1` signal is received, it performs the same operation as the `stat` command.

3. Background

3.1 Web page title

The title of a web page is displayed at the top of the web browser. You can see that the title "성균관대학교" is displayed at the top of the browser for `www.skku.edu` as shown below.

A web page consists of an HTML document, where web page titles are described between `<title>` and `</title>` tags.

3.2 `wget`

- `wget` is a command that receives and stores data from a given URL.
- `wget` Example
  - ✓ The following command saves the HTML document of the web page `http://www.skku.edu` as `1`.

```
// reference: $ man wget
$ wget -O 1 http://www.skku.edu
```

  - ✓ You can see `<title>성균관대학교</title>` in the file `1` as below.

```
4 <!doctype html>
5 <html lang="ko">
6 <head>
7 <title>성 균 관 대 학 교 </title>
8 <link rel="shortcut icon" href="/_res/skku/img/common/favicon.png">
```

3.3 Domain

A domain is a string representation of an address for identifying a device connected to the Internet. It is much easier for us to memorize the address nyx.skku.ac.kr than to memorize the IP address 115.145.179.100. However, domain names must be changed to IP addresses, and this system is called DNS (Domain Name System).

Domain address are interpreted from right to left based on the '.' character.

- The rightmost string is called the TLD (Top Level Domain).
  - ✓ gTLD (generic TLD): Indicates the nature of the web page.
    - ◆ From the third string on the right (towards the left), the sub-domain is managed by the owning organization.
  - ✓ ccTLD (country code TLD): It means each country.
    - ◆ The second string on the right (SLD, Second Level Domain) indicates the nature of the web page.
    - ◆ From the third string on the right, the sub-domain is managed by the owning organization.
- Example 1 (e.g., `http://www.skku.edu/skku/index.do`)
  - ✓ Protocol: `http://`
  - ✓ Sub-domain: `www.skku.edu`
  - ✓ Domain: `skku.edu`
  - ✓ gTLD: `edu`
- Example 2 (e.g., `http://nyx.skku.ac.kr`)
  - ✓ Protocol: `http://`
  - ✓ Sub-domain: `nyx.skku.ac.kr`
  - ✓ Domain: `skku.ac.kr`
  - ✓ ccTLD: `kr`
  - ✓ SLD: `ac`

4. Restrictions

- Work on the assignment by yourself, on a Linux environment.
- Write your name and student ID at the top of the code.
- If a resource is dynamically allocated, it must be released before the program terminates. (e.g., file, memory)
- Assume that you only use **“`.kr`”** as the ccTLD and proceed with the assignment.
- Do NOT use any other libraries other than the standard C library.
  - ✓ If necessary, you can implement it directly in the `collect.c` file.
  - ✓ To perform file I/O, use the following system calls: `open()`, `read()`, `write()`, `close()`, `lseek()`
  - ✓ For process handling, use the following system calls: `fork()`, `exit()`, `wait()`, `exec()` family
  - ✓ For signal handling, use system calls.

5. Hand-in instructions

- Skeleton code consists of:

`Makefile`	File need for GNU make
`collect.c`	Performs collecting web page titles or other commands

- Files to submit: `Makefile`, `collect.c`, `[student_id].pdf (Do not add '[', ']'.)`
  - ✓ To submit, compress the above files into a single file with the name `[student_id].tar.gz` (Do not add '[', ']'.)
  - ✓ To compress the files for submission, use the following command.

```
$ make tar
```

- Write a report describing the implementation method and design for your program in PDF format and submit it as `[student_id].pdf`.
- 10% of your score will be deducted for every 12 hours in case of late submissions.
- In case of plagiarism, you will **FAIL** this course.
  (If two similar source codes or reports are found, both students will fail this course.)
- If you have any questions, you can either reach us through help.eslab@gmail.com or visit room 85465 (C&C Center).

Have fun!

Somm Kim, Sooyun Lee
Sungkyunkwan University, Embedded Software Laboratory