Journal of Engineering and Applied Sciences 13 (Special Issue 1): 2486-2491, 2018
ISSN: 1816-949X

A Study on the Adding a Loop Function to the PSSI Technique

[1]Keun-Young Choi, [2]Hyun-Chang Lee and [2]Kyu-Tae Lee
[1]Department of Information and Communication Engineering,
[2]Division of Information and Communication Engineering, Kongju National University,
31080 Choong-Nam, Cheon-An, Korea

Abstract: In this study, a loop function method that repeatedly performs the same interval is proposed using a simple algorithm for the PSSI technique and the web-document source consisting of external files. The PSSI technique has flexibility of programming and easy modification of web-document source. Once the output format is specified in a web-page that repeats the output of the same format such as BBS (Bulletin Board System) or database, the proposed PSSI loop function uses it repeatedly. Therefore, it is easy to modify the web-document source and reduces the capacity. Also, when the displayed data is finished, the remainder of the results in a web-page is blanked in the conventional PSSI technique. Since, the proposed loop function is displayed up to the data, the appearance of the web-page becomes more proper. In order to verify the effect of the proposed loop function, the PSSI technique program and loop function program were created for BBS program. The performance of the proposed method is evaluated in terms of capacity, maintenance and appearance. Web-pages composed of 15 lines with loop function decreased capacity by 30.57% and modifying time by 1/15 rather than the conventional source document. Also, the web-page with loop function is composed without a blank on any data.

Key words: Dynamic, web, document, PSSI, SSI, CGI

INTRODUCTION

In recent years as IT (Information Technology) devices such as PC (Personal Computer) and smart-phone have been improved dramatically, web-document has been making remarkable progress. In the past, the web-document composed of HTML (Hyper-Text Mark-up Language) was a static document that sent documents from server to user. As a result, information transfer was unilateral from the server to the client. In contrast, nowadays applications of web-document have been diversified by using document techniques which contents are changed according to request of clients (Su and Amin, 2001, Douglas, 1998). CGI (Common Gateway Interface) and SSI (Server-Side Include) (Tian-Huang and Jia Xi, 2007, Kou and Springsteel, 1997, Zhen-Ximg and Xian-Yi, 2008) were proposed communication method for information interchange between user and server by NCSA (National Center for Super-computer Applications). The purpose of communication method is that the server dynamically generates a web-document corresponding to the request of users. SSI is rarely used due to its security drawback. On the other hand, CGI, PHP (Hypertext Preprocessor) and JSP (Java Server Pages) (Brenner and Aoki, 1995) are used for web-document. In Fig. 1, dynamic web-document generation principle of the

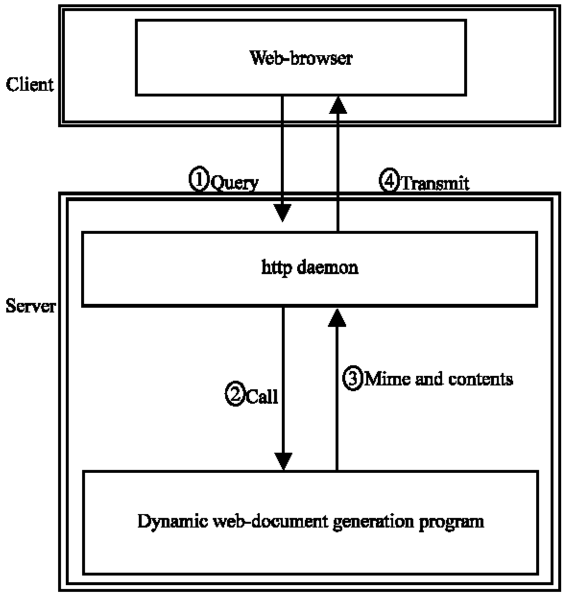


Fig. 1: Flow diagram of generation dynamic web-document by the CGI

Corresponding Author: Hyun-Chang Lee, Division of Information and Communication Engineering, Kongju National University, 31080 Choong-Nam, Cheon-An, Korea

CGI is that documents are generated in the document generation program as documents are requested in the web-browser (Thomas, 1996).

The application of the CGI has been diversified and expanded such as simple visitor counter, information retrieval system, Bulletin Board System (BBS), electronic commerce and web-mail and the research of these fields has been investigated.

Michael established a model for the CGI program and proposed indicators and examples of the programming language selection required for the CGI program (Levy, 1998). Bernhard investigated the functional requirements of the CGI program and the precautions for the program (Wagner, 1998). Also, he proposed the method of indirectly execute a CGI program via. “chuid” program implemented by C-language in order to improve the problem caused by the permission of execution of the CGI program. A script program called ‘WebStory’ using the CGI technique was developed to edit the HTML for connection establishment between WWW-server and enterprise system with WYSWYG (What You See What You Get) (Katoh *et al*., 1996). A study was developed which a formal HTML library function to generate a web-document using the Python language and called it to generate a web-document of the corresponding part (Lee and Lee, 1997). The PSSI (Pseudo-SSI) technique was proposed which has advantages of the CGI that flexibility of programming and security and advantages of the SSI that ease of modification of web-document source. This makes it easy to modify the web-document source of the CGI, reduce program capacity by program integration, and facilitate maintenance and repair. In addition, It proposed a modified PSSI technique for multi-user system which is a program for generating dynamic web-documents and each user can freely generate dynamic web-documents. The PSSI technique is used by deployment of a PSSI tag corresponding to the information module where the web-document source data is to be displayed. Therefore, when a simple dynamic web-document is used, document modification is simple. However, there are some inconveniences that the same parts are repeated such as BBS or document for database output.

In this study, a way to iterate over a specific part of the web-document is proposed by a simple algorithm. The proposed method uses the PSSI technique to fetch a web-document source composed of external file types.

MATERIALS AND METHODS

Adding a loop-function to the PSSI

The PSSI technique: The PSSI (Lee and Lee, 2006) does not include the web-document source in the CGI program but maintains it as an external file. The PSSI inserts the part where variable information is to be included in the web-document source file using a specific tag. The CGI program detects and parses the specific tags and substitutes them with their contents as shown in Fig. 2.

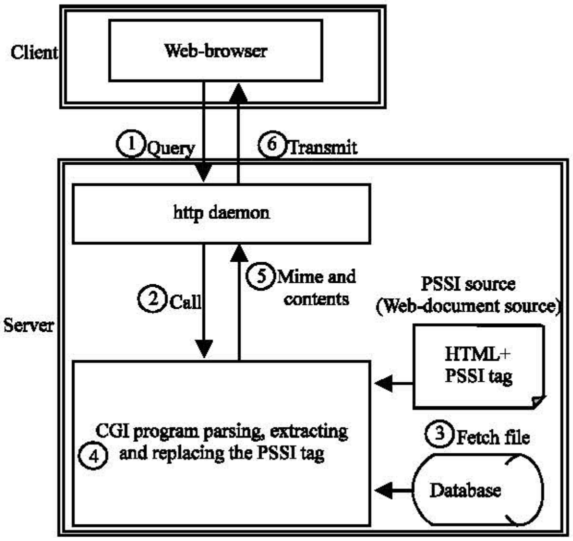


Fig. 2: Flow diagram of the PSSI technique

It is noted that the following sequence:

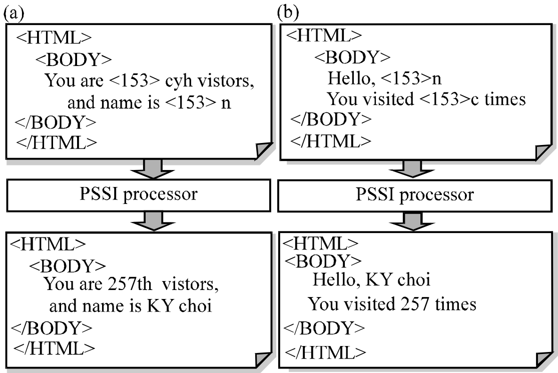
- Web-browser requests service
- The http daemon passes control to the corresponding CGI program
- After, the CGI program internally runs the information preparation process in conjunction with the database, it fetches the web-document source file to generate a dynamic web-document
- It detects and parses the PSSI tag included in the web-document source file and substitutes the tag part with corresponding information contents
- It transmits the completed web-document to the client

The PSSI processing program substitutes the PSSI tag as shown in Fig. 3. In Fig. 3, “<153> c” and “<153> n” in the source document are PSSI tags. The CGI program detects the specific tag “<153>”, parses the code following the tag and substitutes it as follows:

$$<153>c \rightarrow \text{count number}$$

$$<153>n \rightarrow \text{Name}$$

Similar, to the SSI, the PSSI tag is substituted for the information content and sent to the user. Therefore, when the source document is modified, the generated document format can be freely converted. Also, when the PSSI



(a)

```
<HTML>
  <BODY>
    You are <153> cyh vistors,
      and name is <153> n
</BODY>
</HTML>
```

```
<HTML>
  <BODY>
    You are 257th  vistors,
     and name is KY choi
</BODY>
</HTML>
```

(b)

```
<HTML>
   <BODY>
     Hello, <153>n
     You visited <153>c times
</BODY>
</HTML>
```

```
<HTML>
<BODY>
   Hello, KY choi
   You visited 257 times
</BODY>
</HTML>
```

Fig. 3: Examples of substituting the PSSI tag in the CGI program; a) Before modifying and b) After modifying

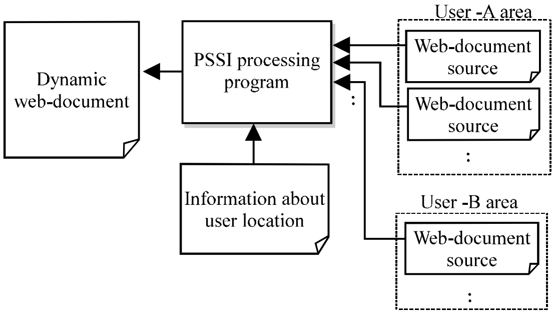
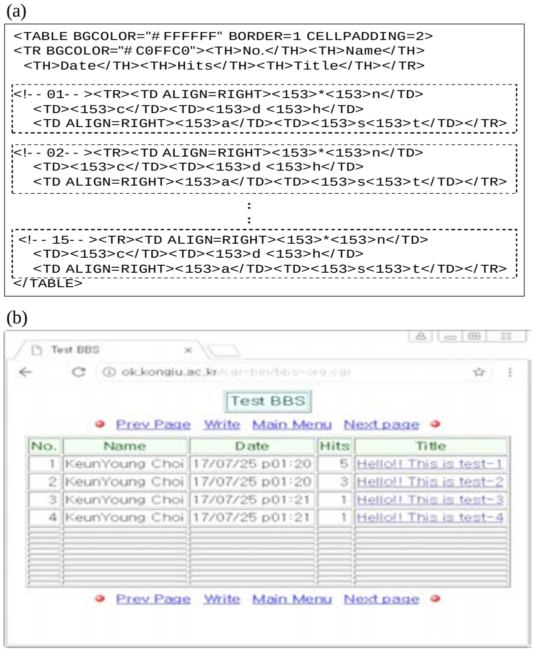


Fig. 4: Example in the multi-user system

technique is applied to a multi-user system by web-document source maintained as an external file, a web-document source is placed in each user directory as shown in Fig. 4. When the PSSI technique fetches the document source in the user directory, a single service program allows the user to freely create and express a desired dynamic web-document for each user. As a result, it has a similar effect as executing individual programs.

It is noted that the PSSI technique has the following characteristics various programming languages can be used to flexibly apply to complex applications, it is general purpose regardless of execution platform, there is no system load due to a separate daemon installation, it is easy to implement and the program becomes simple as the web-document is placed in the external, a program group with a common function can be integrated into one program which makes it proper for a server due to the small program capacity, since, the PSSI tags are treated as information modules, documents that are irrelevant to data order or position can be created when the PSSI technique is applied to a multi-user system each user can freely configure a web-page by modification of the document source in the user's directory, a user can be provided with multiple CGI services with different document formats, there is no need to duplicate programs for each user to create different web-pages which increases efficiency of server management and decreases capacity of program (more than 9 times when using 15 users in the experiment).



```
<TABLE BGCOLOR="# FFFFFF" BORDER=1 CELLPADDING=2>
<TR BGCOLOR="# C0FFC0"><TH>No.</TH><TH>Name</TH>
 <TH>Date</TH><TH>Hits</TH><TH>Title</TH></TR>

<!-- 01-- ><TR><TD ALIGN=RIGHT><153>*<153>n</TD>
  <TD><153>c</TD><TD><153>d <153>h</TD>
  <TD ALIGN=RIGHT><153>a</TD><TD><153>s<153>t</TD></TR>

<!-- 02-- ><TR><TD ALIGN=RIGHT><153>*<153>n</TD>
  <TD><153>c</TD><TD><153>d <153>h</TD>
  <TD ALIGN=RIGHT><153>a</TD><TD><153>s<153>t</TD></TR>
                          :
                          :
 <!-- 15-- ><TR><TD ALIGN=RIGHT><153>*<153>n</TD>
  <TD><153>c</TD><TD><153>d <153>h</TD>
  <TD ALIGN=RIGHT><153>a</TD><TD><153>s<153>t</TD></TR>
</TABLE>
```

No.	Name	Date	Hits	Title
1	KeunYoung Choi	17/07/25 p01:20	5	Hello!! This is test-1
2	KeunYoung Choi	17/07/25 p01:20	3	Hello!! This is test-2
3	KeunYoung Choi	17/07/25 p01:21	1	Hello!! This is test-3
4	KeunYoung Choi	17/07/25 p01:21	1	Hello!! This is test-4

Fig. 5: Weakness example of the PSSI technique; a) Repeated part of the web-document source and b) Data ended table

Limitation of the PSSI: Since, the PSSI technique deploys the PSSI tag corresponding to the information module in the part where the data of a web-document source is displayed, it is easy to modify the dynamic web-document. However, there are some inconveniences that repetition of the same parts such as a BBS or database output document. First, since, the content of the web-document source is repeatedly listed in the table to be repeated, the capacity increases as shown in Fig. 5a. Also, in order to modify the contents of the repeat part, it is necessary to modify all the repeatedly listed sentences, which is inefficient and can cause a mistake. Second, when the data displayed on a web-page using table is finished but the table part of the web-document source still remains, the rest of the table is blanked and disfigured the web-page as shown in Fig. 5b. Thus, if the repeated lines are processed by the same loop function as that used in the general language, the source document becomes concise, the editing becomes easy and the modification time can be shortened.

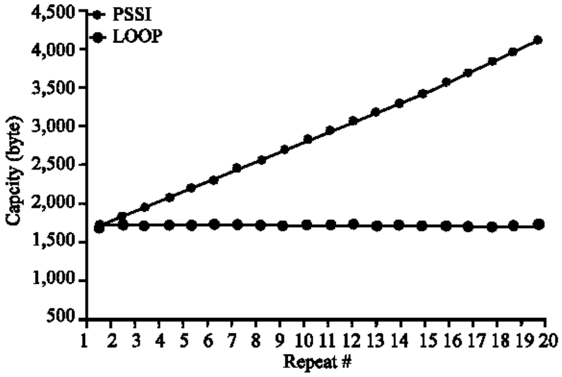


Fig. 6: Simulation result of an example web-document

In addition, since, it is possible to generate the necessary data for repeated data, the blank part of the table can be eliminated and the disfigurement problem of the web-page can be solved. The required capacity to create a repeated document using the PSSI technique can be expressed as:

$$C_{PSSI} = C_{rpt} \times n + C_{etc} \qquad (1)$$

Where:

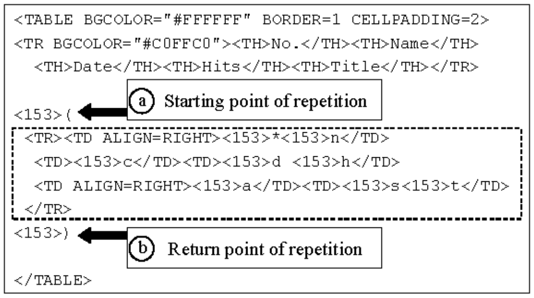
C_{rpt} = The Capacity of the part to be repeated
C_{etc} = The Capacity of remaining part

The capacity required to create a document using a repeat tag can be expressed as:

$$C_{Loop} = C_{rpt} + C_{ctl} + C_{etc} \qquad (2)$$

Where, C_{ctl} is the repeat control statement. For example, C_{rpt} = 142, C_{etc} = 1,230, C_{ctl} = 12 as shown in Fig. 5a and 6 shows the simulation of the capacity for the number of iteration. Consequently, while the loop function maintains the capacity regardless of the number of repeats, the document capacity that does not use the loop function is proportional to the number of repeats.

Realization of the PSSI loop function: Figure 7 shows the principle of the PSSI loop function and Fig. 8 shows the flowchart of the PSSI loop function. In Fig. 7, “<153> (“is the PSSI tag which repetition starts and “<153>)” is the PSSI tag which repetition ends. The PSSI processing program detects the PSSI tag while fetching the web-document source file. When parsing the tag code and replacing it with the contents, if the repeated start tag is parsed as in point (a) as shown in Fig. 7, the file pointer (i.e., the file offset after “<153>(") is stored in the variable. If two or more loops are to be configured, each loop level can have its own tag, or a stack can be configured to push and pop file pointers.



```
<TABLE BGCOLOR="#FFFFFF" BORDER=1 CELLPADDING=2>
<TR BGCOLOR="#C0FFC0"><TH>No.</TH><TH>Name</TH>
  <TH>Date</TH><TH>Hits</TH><TH>Title</TH></TR>
<153>(
 <TR><TD ALIGN=RIGHT><153>*<153>n</TD>
  <TD><153>c</TD><TD><153>d <153>h</TD>
  <TD ALIGN=RIGHT><153>a</TD><TD><153>s<153>t</TD>
 </TR>
<153>)
</TABLE>
```

a Starting point of repetition
b Return point of repetition

Fig. 7: Principle of the PSSI loop function

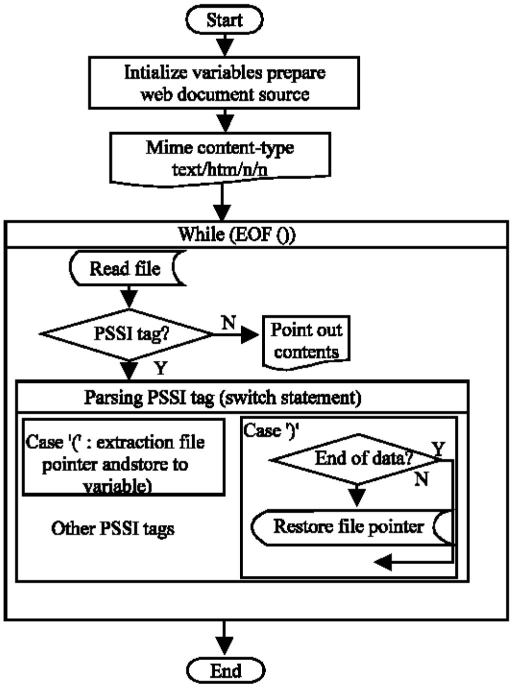
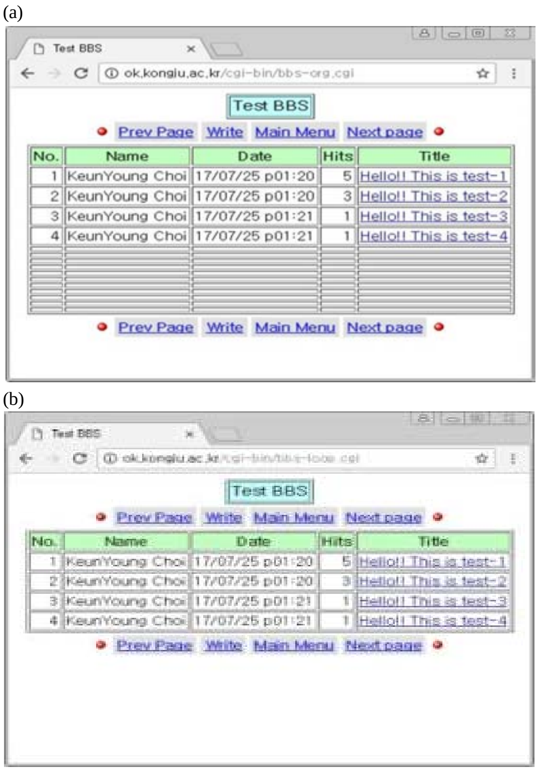

Start
Intialize variables prepare web document source
Mime content-type text/htm/n/n
While (EOF ())
Read file
PSSI tag?
N
Point out contents
Y
Parsing PSSI tag (switch statement)
Case '(' : extraction file pointer andstore to variable)
Other PSSI tags
Case ')'
End of data?
Y
N
Restore file pointer
End

Fig. 8: Flowchart for processing the PSSI loop function

When the fetch of the source file proceeds and the repetition end tag is parsed in the point (b), it is checked whether the record or data of database is terminated. If terminated, the tag is ignored and preceded. If records or data remain, they return to the file pointer after the point (a) by using the fseek() function and it is repeated between the point (a) and point (b).

RESULTS AND DISCUSSION

Experiments and considerations: In order to verify the effect of the proposed loop function, a simple BBS program was created using the proposed algorithm as shown in Fig. 7 and 8 and the results are shown in Fig. 9. Figure 9a shows execution of the test BBS program using general PSSI. If there are only four records in a web-page



(a)

Test BBS

ok,kongiu,ac,kr/cgi-bin/bbs-org,cgi

Test BBS

● Prev Page Write Main Menu Next page ●

No.	Name	Date	Hits	Title
1	KeunYoung Choi	17/07/25 p01:20	5	Hello!! This is test-1
2	KeunYoung Choi	17/07/25 p01:20	3	Hello!! This is test-2
3	KeunYoung Choi	17/07/25 p01:21	1	Hello!! This is test-3
4	KeunYoung Choi	17/07/25 p01:21	1	Hello!! This is test-4

● Prev Page Write Main Menu Next page ●

(b)

Test BBS

ok,kongiu,ac,kr/cgi-bin/bbs-loop,cgi

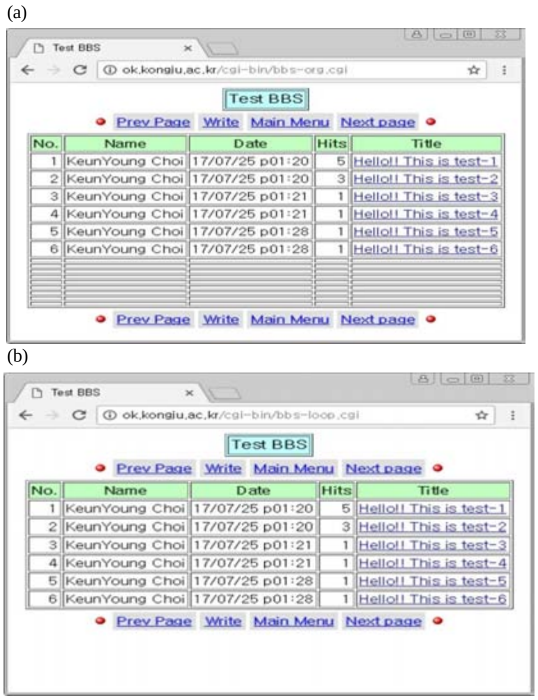
Test BBS

● Prev Page Write Main Menu Next page ●

No.	Name	Date	Hits	Title
1	KeunYoung Choi	17/07/25 p01:20	5	Hello!! This is test-1
2	KeunYoung Choi	17/07/25 p01:20	3	Hello!! This is test-2
3	KeunYoung Choi	17/07/25 p01:21	1	Hello!! This is test-3
4	KeunYoung Choi	17/07/25 p01:21	1	Hello!! This is test-4

● Prev Page Write Main Menu Next page ●

Fig. 9: Screen shot of web-pages for experiment; a) PSSI and b) loop function



(a)

Test BBS

ok,kongiu,ac,kr/cgi-bin/bbs-org,cgi

Test BBS

● Prev Page Write Main Menu Next page ●

No.	Name	Date	Hits	Title
1	KeunYoung Choi	17/07/25 p01:20	5	Hello!! This is test-1
2	KeunYoung Choi	17/07/25 p01:20	3	Hello!! This is test-2
3	KeunYoung Choi	17/07/25 p01:21	1	Hello!! This is test-3
4	KeunYoung Choi	17/07/25 p01:21	1	Hello!! This is test-4
5	KeunYoung Choi	17/07/25 p01:28	1	Hello!! This is test-5
6	KeunYoung Choi	17/07/25 p01:28	1	Hello!! This is test-6

● Prev Page Write Main Menu Next page ●

(b)

Test BBS

ok,kongiu,ac,kr/cgi-bin/bbs-loop,cgi

Test BBS

● Prev Page Write Main Menu Next page ●

No.	Name	Date	Hits	Title
1	KeunYoung Choi	17/07/25 p01:20	5	Hello!! This is test-1
2	KeunYoung Choi	17/07/25 p01:20	3	Hello!! This is test-2
3	KeunYoung Choi	17/07/25 p01:21	1	Hello!! This is test-3
4	KeunYoung Choi	17/07/25 p01:21	1	Hello!! This is test-4
5	KeunYoung Choi	17/07/25 p01:28	1	Hello!! This is test-5
6	KeunYoung Choi	17/07/25 p01:28	1	Hello!! This is test-6

● Prev Page Write Main Menu Next page ●

Fig. 10: When the additional records added; a) PSSI and b) Loop function

configured to display 15 records per page, all remaining lines blanked that web-page is disfigured as shown in Fig. 9 b. In contrast, the record display line appears when there is data using the loop function. The line is not shown where the data is not filled, so that, unnecessary parts are removed as shown in Fig. 9 b. Two records are added under the same condition as shown in Fig. 10 a b, respectively. The general PSSI shows that the blank line is reduced but still remains as shown in Fig. 10 a. When the loop function is used, the necessary lines are generated as shown in Fig. 10 b.

The capacity of the PSSI source file used in Fig. 9 and 10 is 2,855 bytes for general PSSI whereas the capacity of loop function is 930 bytes which is reduced to 30.57%. Also, when user modifies the PSSI source file, the source file for general PSSI should be modified for all of the same lines. Since, only one line needs to be modified for the loop function, the modification time is also greatly reduced by the number of lines.

CONCLUSION

In this study, a loop function method that repeatedly performs the same interval is proposed using a simple algorithm for the PSSI technique and the web-document source consisting of external files. The PSSI technique has flexibility of programming and easy modification of web-document source. Once the output format is specified in a web-page that repeats the output of the same format such as BBS (bulletin board system) or database, the proposed PSSI loop function uses it repeatedly. Therefore, it is easy to modify the web-document source and reduces the capacity. Also, when the displayed data is finished, the remainder of the results in a web-page is blanked in the conventional PSSI technique. Since, the proposed loop function is displayed up to th data, the appearance of the web-page becomes more proper. In order to verify the effect of the proposed loop function, the existing PSSI technique program and loop function program were created for BBS program. Then, it was installed and run on the web-server. Web-pages composed of 15 lines with loop function decreased capacity by 30.57% and modifying time by 1/15 rather than the conventional source document. Also, the web-page with loop function is composed without a blank on any data.

REFERENCES

Brenner, S.E. and E. Aoki, 1995. Introduction to CGI-PERL: Getting Started with Web Scripts. 1st Edn., IDG Books Worldwide, Foster, California, USA.,.

Douglas, E.C., 1998. Computer Networks and Internets. 2nd Edn., Prentice-Hall, Upper Saddle River, New Jersey, USA., ISBN:9780135906545, Pages: 506.

HyeonJong, K. and L. KiHo, 1997. Design and implementation of Web documents creating system with HTML-supporting library. J. KIISE. Comput. Practices Lett., 3: 375-383.

Katoh, T., N. Nakagoshi and M. Yamazaki, 1996. HTML editor web story supporting cooperation between WWW server and enterprise system. NEC. Tech. J., 49: 157-161.

Kou, C. and F. Springsteel, 1997. The security mechanism in the World Wide Web (WWW) and the Common Gateway Interface (CGI): Example of central police university entrance examination system. Proceedings of the 31st IEEE Annual International Carnahan Conference on Security Technology, October 15-17, 1997, IEEE, Canberra, Australia, pp: 114-119.

Lee, H.C. and J.E. Lee, 2006. A study on the effective method of generating the dynamic web documents in the multi-user system. J. Korean Inst. Commun. Inf. Sci., 31: 478-485.

Levy, M.R., 1998. Web programming in guide. Software Pract. Experience, 28: 1581-1603.

Su, D. and N. Amin, 2001. A CGI-based approach for remotely executing a large program for integration of design and manufacture over the Internet. Intl. J. Comput. Integr. Manuf., 14: 55-65.

Thomas, B., 1996. CGI Programming in C and Perl. Addison-Wesley, Boston, Massachusetts, USA., ISBN:9780201422191, Pages: 405.

Tian-Huang, C. and H. Jia-Xi, 2007. Design and realization of CGI in embedded dynamic web technology. Proceedings of the IFIP International Conference on Network and Parallel Computing Workshops NPC, September 18-21, 2007, IEEE, Liaoning, China, pp: 774-777.

Wagner, B., 1998. Controlling CGI programs. ACM. SIGOPS Operating Syst. Rev., 32: 40-46.

Zhen-Xing, W. and R. Xian-Yi, 2008. A study on CGI of embedded webserver. Proceedings of the International Symposium on Computer Science and Computational Technology Vol. 1, December 20-22, 2008, IEEE, Shanghai, China, ISBN:978-0-7695-3498-5, pp: 480-483.