

2023년 1학기 시스템프로그래밍 & 시스템 프로그래밍 실습

Assignment 3-3

System Software Laboratory
College of Software and Convergence
Kwangwoon Univ.

Assignment 3-3

- **Process 관리 및 client 접속 정보를 log 파일에 기록하는 프로그램 작성**
 - Assignment 3-2 내용을 유지 하면서 아래에 추가된 내용을 콘솔에 출력 (Execution Example)
 - ① Client가 요청한 경로 (URL부분) (e.g. */hello*)
 - ② 연결 지속 시간
 - 클라이언트의 연결이 disconnected 될 때 출력
 - 단위: 마이크로 초(us)
 - **클라이언트 연결 종료 시간 까지만 출력!**
 - 즉, `sleep(5);` 로 소요된 시간은 제외
 - 로그 파일(*server_log.txt*)에 추가로 작성
 - Log 작성 시, **child 프로세스마다 생성한 thread를 이용**하여 기록
 - 즉, thread 구동 함수가 string을 parameter로 받아 이를 file에 기록하게끔 구현

Assignment 3-3 (cont'd)

- **Process 관리 및 client 접속 정보를 log 파일에 기록하는 프로그램 작성 (cont'd)**
 - 공유 자원인 **로그 파일(server_log.txt) 동기화를 semaphore를 이용하여 보장**
 - Semaphore를 사용하여 file에 접근 부분을 critical section으로 설정
 - 한 번에 하나의 thread만 접근 가능하도록 보장
 - Pre-forked server에 클라이언트가 연결되면, 접속 정보를 로그 파일(server_log.txt)에 작성
 - 로그파일에는 history 기록하지 않음.

Log file example

```
[[Fri May 31 12:58:49 2019] Server is started.  
[Fri May 31 12:58:49 2019] 7786 process is forked.  
[Fri May 31 12:58:49 2019] idleProcessCount : 1  
[Fri May 31 12:58:49 2019] 7788 process is forked.  
[Fri May 31 12:58:49 2019] idleProcessCount : 2  
[Fri May 31 12:58:49 2019] 7790 process is forked.  
[Fri May 31 12:58:49 2019] idleProcessCount : 3  
[Fri May 31 12:58:49 2019] 7792 process is forked.  
[Fri May 31 12:58:49 2019] idleProcessCount : 4  
[Fri May 31 12:58:49 2019] 7794 process is forked.  
[Fri May 31 12:58:49 2019] idleProcessCount : 5
```

```
===== New Client =====
```

```
TIME : [Fri May 31 12:58:56 2019]
```

```
URL : /hello
```

```
IP : 127.0.0.1
```

```
Port : 7786
```

```
PID : 16585
```

```
=====
```

```
[Fri May 31 12:58:56 2019] idleProcessCount : 4
```

```
===== New Client =====
```

```
TIME : [Fri May 31 12:58:58 2019]
```

```
URL : /hello
```

```
IP : 127.0.0.1
```

```
Port : 7788
```

```
PID : 17097
```

```
=====
```

Log file example

```
[Fri May 31 12:58:58 2019] idleProcessCount : 3  
[Fri May 31 12:58:58 2019] 8103 process is forked.  
[Fri May 31 12:58:58 2019] idleProcessCount : 4  
[Fri May 31 12:58:58 2019] 8105 process is forked.  
[Fri May 31 12:58:58 2019] idleProcessCount : 5
```

```
===== Disconnected client =====
```

```
TIME : [Fri May 31 12:59:01 2019]
```

```
URL : /hello
```

```
IP : 127.0.0.1
```

```
Port : 16585
```

```
PID : 7786
```

```
CONNECTING TIME: 7589(us)
```

```
=====
```

```
[Fri May 31 12:59:01 2019] idleProcessCount : 6
```

```
===== Disconnected client =====
```

```
TIME : [Fri May 31 12:59:03 2019]
```

```
URL : /hello
```

```
IP : 127.0.0.1
```

```
Port : 17097
```

```
PID : 7788
```

```
CONNECTING TIME: 8124(us)
```

```
=====
```

①

②

①

②

Log file example

```
[Fri May 31 12:59:03 2019] idleProcessCount : 7  
[Fri May 31 12:59:03 2019] 8105 process is terminated.  
[Fri May 31 12:59:03 2019] idleProcessCount : 6  
[Fri May 31 12:59:03 2019] 8103 process is terminated.  
[Fri May 31 12:59:03 2019] idleProcessCount : 5
```

```
===== New Client =====
```

```
TIME : [Fri May 31 12:59:04 2019]
```

```
URL : /hello
```

```
IP : 127.0.0.1
```

```
Port : 7790
```

```
PID : 17609
```

```
=====
```

```
[Fri May 31 12:59:04 2019] idleProcessCount : 4
```

①

Log file example

===== Disconnected client =====

TIME : [Fri May 31 12:59:09 2019]

URL : /hello

IP : 127.0.0.1

Port : 17609

PID : 7790

CONNECTING TIME: 187272(us)

[Fri May 31 12:59:09 2019] idleProcessCount : 5

^c

[Fri May 31 12:59:15 2019] 7794 process is terminated.

[Fri May 31 12:59:15 2019] idleProcessCount : 4

[Fri May 31 12:59:15 2019] 7792 process is terminated.

[Fri May 31 12:59:15 2019] idleProcessCount : 3

[Fri May 31 12:59:15 2019] 7790 process is terminated.

[Fri May 31 12:59:15 2019] idleProcessCount : 2

[Fri May 31 12:59:15 2019] 7788 process is terminated.

[Fri May 31 12:59:15 2019] idleProcessCount : 1

[Fri May 31 12:59:15 2019] 7786 process is terminated.

[Fri May 31 12:59:15 2019] idleProcessCount : 0

Execution example

```
server_log.txt x
1 [Fri May 31 12:58:49 2019] Server is started.
2 [Fri May 31 12:58:49 2019] 7786 process is forked.
3 [Fri May 31 12:58:49 2019] idleProcessCount : 1
4 [Fri May 31 12:58:49 2019] 7788 process is forked.
5 [Fri May 31 12:58:49 2019] idleProcessCount : 2
6 [Fri May 31 12:58:49 2019] 7790 process is forked.
7 [Fri May 31 12:58:49 2019] idleProcessCount : 3
8 [Fri May 31 12:58:49 2019] 7792 process is forked.
9 [Fri May 31 12:58:49 2019] idleProcessCount : 4
10 [Fri May 31 12:58:49 2019] 7794 process is forked.
11 [Fri May 31 12:58:49 2019] idleProcessCount : 5
12
13 ===== New Client =====
14 TIME : [Fri May 31 12:58:56 2019]
15 URL : /Hello
16 IP : 127.0.0.1
17 Port : 7786
18 PID : 16585
19
20
21 [Fri May 31 12:58:56 2019] idleProcessCount : 4
22
23 ===== New Client =====
24 TIME : [Fri May 31 12:58:58 2019]
25 URL : /Hello
26 IP : 127.0.0.1
27 Port : 7788
28 PID : 17097
29
30
31 [Fri May 31 12:58:58 2019] idleProcessCount : 3
32 [Fri May 31 12:58:58 2019] 8103 process is forked.
33 [Fri May 31 12:58:58 2019] idleProcessCount : 4
34 [Fri May 31 12:58:58 2019] 8105 process is forked.
35 [Fri May 31 12:58:58 2019] idleProcessCount : 5
36
37 ===== Disconnected client =====
38 TIME : [Fri May 31 12:59:01 2019]
39 URL : /Hello
40 IP : 127.0.0.1
41 Port : 16585
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
sslab@sslab:~/projects/ass4-2_v2/Test/Git/System-Programming-Http$ ./semaphore server 39998
[Fri May 31 12:58:49 2019] Server is started.
[Fri May 31 12:58:49 2019] 7786 process is forked.
[Fri May 31 12:58:49 2019] idleProcessCount : 1
[Fri May 31 12:58:49 2019] 7788 process is forked.
[Fri May 31 12:58:49 2019] idleProcessCount : 2
[Fri May 31 12:58:49 2019] 7790 process is forked.
[Fri May 31 12:58:49 2019] idleProcessCount : 3
[Fri May 31 12:58:49 2019] 7792 process is forked.
[Fri May 31 12:58:49 2019] idleProcessCount : 4
[Fri May 31 12:58:49 2019] 7794 process is forked.
[Fri May 31 12:58:49 2019] idleProcessCount : 5

===== New Client =====
TIME : [Fri May 31 12:58:56 2019]
URL : /Hello
IP : 127.0.0.1
Port : 16585
PID : 7786

[Fri May 31 12:58:56 2019] idleProcessCount : 4

===== New Client =====
TIME : [Fri May 31 12:58:58 2019]
URL : /Hello
IP : 127.0.0.1
Port : 17097
PID : 7788

[Fri May 31 12:58:58 2019] idleProcessCount : 3
[Fri May 31 12:58:58 2019] 8103 process is forked.
[Fri May 31 12:58:58 2019] idleProcessCount : 4
[Fri May 31 12:58:58 2019] 8105 process is forked.
[Fri May 31 12:58:58 2019] idleProcessCount : 5

===== Connection History =====
No. IP PID PORT TIME
1 127.0.0.1 7788 17097 Fri May 31 12:58:58 2019
1 127.0.0.1 7786 16585 Fri May 31 12:58:56 2019

===== Disconnected client =====
TIME : [Fri May 31 12:59:01 2019]
URL : /Hello
IP : 127.0.0.1
Port : 16585
PID : 7786
CONNECTING TIME: 7589(us)

[Fri May 31 12:59:01 2019] idleProcessCount : 6
```

1: semaphore_server_3

/home/sslab/projects/ass4-2_v2/Test/Git/System-Programming-Http -

/home/sslab/projects/ass4-2_v2/Test/Git/System-Programming-Http -

127.0.0.1:39998/Hello 90%

System Programming Http

Directory path : /home/sslab/projects/ass4-2_v2/Test/Git/System-Programming-Http-Hello

total 8

Name	Permission	Link	Owner	Group	Size	Last Modified
2	drwxrwxr-x	2	sslab	sslab	4096	May 30 18:23
5	drwxrwxr-x	5	sslab	sslab	4096	May 31 12:50

로그 파일에 history 기록 하지 않음.

Assignment 3-3

- **Code requirements**

- 이전 과제 부분에 문제가 있는 경우 감점
- 출력 형식에 맞지 않으면 감점
- **포트 번호 40000 으로 고정**

- **Makefile requirements**

- 실행 파일이 “semaphore_server”로 생성되도록 Makefile 작성
- 컴파일 도중 warning 발생 시 감점
- **“\$ make” 를 통해 실행 파일이 생성되지 않는 경우, 0점**

Report Requirements

- **Ubuntu 16.04.5 Desktop 64bits 환경에서 채점**
- **Copy 발견 시 0점 처리**
- **보고서 구성**
 - **보고서 표지**
 - 수업 명, 과제 이름, 담당 교수님, 학번, 이름 필히 명시
 - 과제 이름 → assignment3-3
 - **과제 내용**
 - Introduction)
 - 과제 소개 - 4줄 이상(background 제외) 작성
 - Flow chart (appendix 참고)
 - Pseudo code (appendix 참고)
 - 결과화면
 - 수행한 내용을 캡처 및 설명
 - 고찰
 - 과제를 수행 결과 분석 및 해석 해결한 부분, 문제점 발생에 대한 요약
 - Reference
 - 과제를 수행하면서 참고한 내용을 구체적으로 기록
 - 강의자료만 이용한 경우 생략 가능

Report Requirements

- Softcopy Upload

- 제출 파일

- 보고서 + 소스파일 하나의 압축 파일로 압축하여 제출(tar.gz)
 - 보고서(.pdf. 파일 변환)
 - 소스코드
 - 학번_semaphore_server.c
 - Makefile
 - [httpd.conf]
 - 실행 파일명 : semaphore_server
 - 실행 파일명 다르게 작성 시 감점

- 보고서 및 압축 파일 명 양식

- Web3_3_수강분류코드_학번 으로 작성

수강요일	이론1 월5수6	이론2 월2수1	실습1 금12	실습2 금56	실습3 금78
수강분류 코드	A	B	C	D	E


- 예시-이론 월5 수6 수강하는 학생인경우

- 보고서 Web3_3_A_20231234567.pdf

압축 파일 명: Web3_3_A_20231234567.tar.gz

Report Requirements

- 실습 수업을 수강하는 학생인 경우
 - 실습 과목에 과제를 제출(.tar.gz)
 - 이론 과목에 간단한 .txt 파일로 제출

 실습수업때제출했습니다.

2022-08-29 오후 3:58

텍스트 문서

0KB

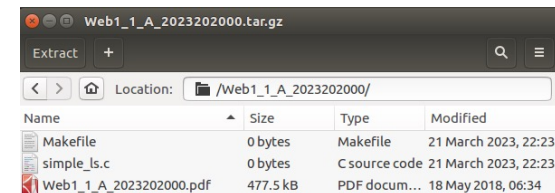
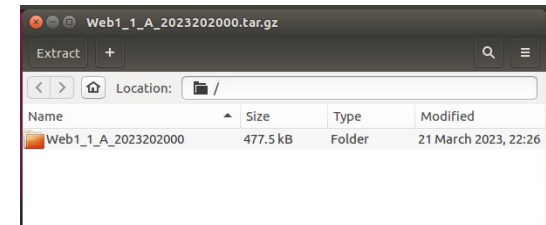
- 이론 과목에 .txt 파일 미 제출 시 감점

- 과제 제출
 - **KLAS – 강의 과제 제출**
 - **2023년 5월 31일 수요일 23:59까지 제출**
 - **딜레이 받지 않음**
 - 제출 마감 시간 내 미제출시 해당 과제 0점 처리
 - 교내 서버 문제 발생 시, 메일로 과제 제출 허용
- **Tar 압축 및 해제 방법**
 - 압축 시 → `tar -zcvf [압축 파일명].tar.gz[폴더 명]`
 - 해제 시 → `tar -zxvf 파일명.tar.gz`

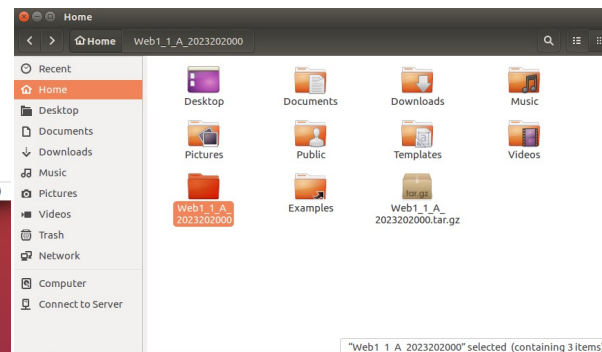
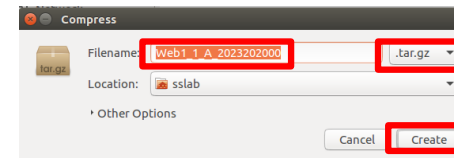
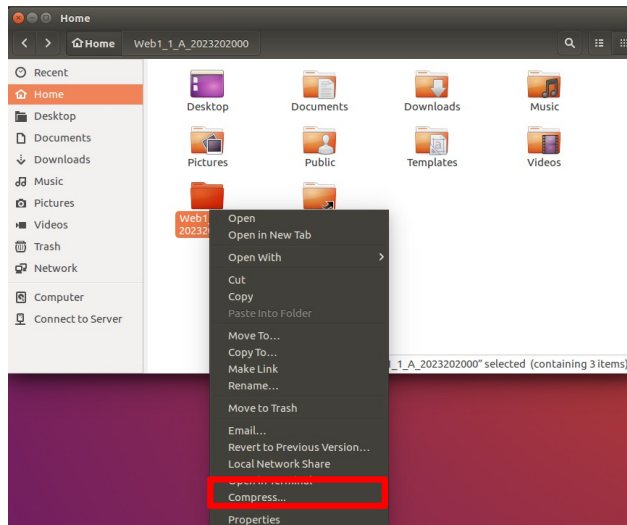
과제 제출 양식

- 소스파일, 보고서, Makefile만 하나의 tar.gz 로 압축, 아래 두 가지 방법 중 택 1
- 압축 명령어 사용

```
sslslab@ubuntu: ~/Web1_1_A_2023202000
sslslab@ubuntu:~$ ls
Desktop  Downloads  Music  Public  Videos
Documents examples.desktop Pictures Templates Web1_1_A_2023202000
sslslab@ubuntu:~$ cd Web1_1_A_2023202000/
sslslab@ubuntu:~/Web1_1_A_2023202000$ ls
Makefile simple_ls.c Web1_1_A_2023202000.pdf
sslslab@ubuntu:~/Web1_1_A_2023202000$
sslslab@ubuntu:~$ tar -zcvf Web1_1_A_2023202000.tar.gz Web1_1_A_2023202000/
Web1_1_A_2023202000/
Web1_1_A_2023202000/Makefile
Web1_1_A_2023202000/simple_ls.c
Web1_1_A_2023202000/Web1_1_A_2023202000.pdf
sslslab@ubuntu:~$ ls
Desktop  examples.desktop  Public  Web1_1_A_2023202000
Documents  Music  Templates  Web1_1_A_2023202000.tar.gz
Downloads  Pictures  Videos
sslslab@ubuntu:~$
```



- 디렉토리에서 직접 압축



2023년 1학기 시스템프로그래밍실습

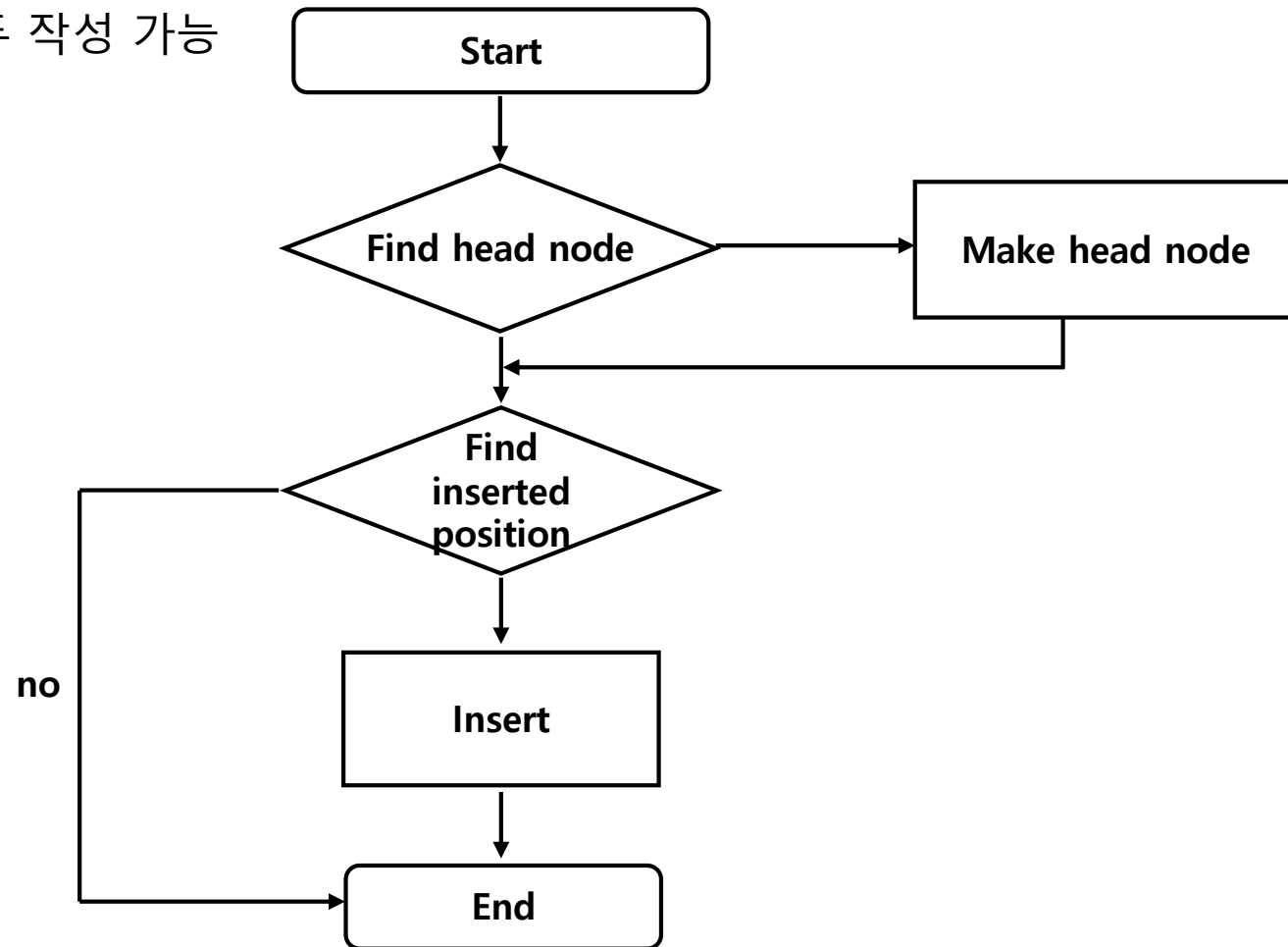
Appendix

System Software Laboratory
College of Software and Convergence
Kwangwoon Univ.

보고서 작성 요령 (1/2)

■ Algorithm – Flow Chart (Each function)

- E.g.
- 국문, 영문 모두 작성 가능



보고서 작성 요령 (2/2)

■ Algorithm – Pseudo Code

- E.g.
- 국문, 영문 모두 작성 가능

```
FixHeap(Node *root, Key k)
{
    Node vacant, largerChild;
    vacant = root;
    while( vacant is not leaf ) {
        largerChild = the child of vacant with the larger key;
        if( k < largerChild's Key ) {
            copy largerChild's key to vacant;
            vacant = largerChild;
        }
        else exit loop;
    }
}
```

```
Set grade counter to one
While grade counter is less than or equal to ten
    Input the next grade
    Add the grade into the total
Set the class average to the total divided by ten
Print the class average
```


Comment 작성 요령 (1/3)

■ File Head Comment

- E.g.
- 국문, 영문 모두 작성 가능

```
////////////////////////////////////  
// File Name      : Main.c                               //  
// Date          : 2023/03/01                             //  
// Os            : Ubuntu 16.04 LTS 64bits                //  
// Author        : Hong Gil Dong                         //  
// Student ID    : 2023123456                             //  
// ----- //  
// Title : System Programming Assignment #1-1 (proxy server) //  
// Description : ...                                       //  
////////////////////////////////////
```

Comment 작성 요령 (2/3)

■ Function Head Comment

- E.g.
- 국문, 영문 모두 작성 가능

```
////////////////////////////////////  
// InsertNode //  
// ===== //  
// Input: Node* -> Insert Node, //  
//         Node* -> Column node before insert node //  
//         Node* -> Row node before insert node //  
//         (Input parameter Description) //  
// Output: int - 1 success //  
//           0 fail //  
//         (Out parameter Description) //  
// Purpose: Inserting node //  
////////////////////////////////////
```

Comment 작성 요령 (3/3)

■ In-line Comment

- e.g.
- 국문, 영문 모두 작성 가능

```
if( pRowPos->pNextRow != pRowPos ) {  
    pTemp->pNextRow = pRowPos->pNextRow;          // pTemp set next row  
    if( !( pRowPos->pNextRow->bHead ) ){  
        pRowPos->pNextRow->NodeItem.pPrevRow = pTemp;  
    } // end of if  
} // end of if  
else {  
    pTemp->pNextRow = pRowPos;                      // pTemp set next row  
} // end of else  
pTemp->NodeItem.pPrevRow = pRowPos;                // pTemp set previous row  
pRowPos->pNextRow = pTemp;
```