

2023년 1학기 시스템프로그래밍 & 시스템 프로그래밍 실습

Assignment 3-1

System Software Laboratory
College of Software and Convergence
Kwangwoon Univ.

Assignment 3-1

- Assignment 2-3 에 다음 사항을 추가
 - Makefile 작성 시
 - **Preforked_server** 으로 실행파일 생성.
 - Pre-forked 방식으로 server 구현
 - 5개 child 프로세스를 생성 및 유지
 - 부모 프로세스는 fork()로 자식 프로세스 만들고, terminal에 그 정보를 출력 (“... forked.”)
 - SIGINT 발생 시, 모든 프로세스를 완전하게 종료 시켜야 함
 - 프로세스가 수행해야 할 연산
 - 부모 프로세스
 - (1) 자식 프로세스 상태 출력 (“... terminated.”)
 - (2) 해당 프로세스 상태 출력 (“... terminated.”)
 - (3) 종료
 - 자식 프로세스: 종료
 - 단, zombie 프로세스가 생성되지 않도록 보장해야 함
 - Zombie 프로세스 확인: ps [pid] -> STAT에 z 출력 시, zombie 프로세스

Assignment 3-1

- Assignment 2-3 에 다음 사항을 추가 (cont'd)
 - Connection history 출력 (과제 2-3의 연장)
 - 10초마다 자식 프로세스에 연결된 client의 접속 기록(connection history)를 출력해야 함
 - 각 자식 프로세스 마다 최신 history 기록을 저장 및 출력 (자식 프로세스 당 최대 10개)
 - 즉, 5개의 자식 프로세스의 전체 history 개수는 최대 50개
 - History 번호 (“No.”)는 자식 프로세스 마다 개별적으로 유지
 - History 제목은 부모 프로세스가 출력, 하위에 있는 내용은 자식 프로세스가 출력
 - 즉, “... Connection history ...”, “No.”, “IP” 등의 제목 부분은 부모 프로세스에서 출력
 - 부모 프로세스가 history 제목을 출력한 뒤, SIGUSR1 signal을 자식 프로세스에게 전달하여 history 내용 출력을 지시
 - 본 과제에서는 history 내용 출력 관련한 동기화 문제를 고려하지 않음
 - 단, 2-3에서 출력하였던 request 수는 본 과제에서 출력하지 않음
 - 터미널에 간단한 로그 기록을 출력
 - “... Server is started.”
 - “... Server is terminated”

Assignment 3-1 (cont'd)

결과화면

```
[Wed May 15 18:00:00 2019] Server is started.
[Wed May 15 18:00:00 2019] 2332 process is forked.
[Wed May 15 18:00:00 2019] 2333 process is forked.
[Wed May 15 18:00:00 2019] 2334 process is forked.
[Wed May 15 18:00:00 2019] 2335 process is forked.
[Wed May 15 18:00:00 2019] 2336 process is forked.
===== New client =====
[Wed May 15 18:05:34 2019]
IP : 223.195.33.43
Port : 22343
=====
===== Connection History =====
NO.      IP          PID      PORT      TIME
1        223.195.33.43  2333     22343     Wed May 15 18:05:34 2019

===== New client =====
[Wed May 15 18:07:35 2019]
IP : 223.222.23.55
Port : 22333
=====
===== Connection History =====
NO.      IP          PID      PORT      TIME
1        223.195.33.43  2333     22343     Wed May 15 18:05:34 2019
1        223.222.23.55  2335     22333     Wed May 15 18:07:35 2019
```

Server program 시작

Child process 생성 시

Client 접속 시

History 제목 → 부모 프로세스가 출력
History 내용 → 자식 프로세스가 출력

Client 접속 시

History 제목 → 부모 프로세스가 출력
History 내용 → 자식 프로세스가 출력

```
===== Connection History =====
NO.      IP          PID      PORT      TIME
1        223.195.33.43  2333     22343     Wed May 15 18:05:34 2019
1        223.222.23.55  2335     22333     Wed May 15 18:07:35 2019
```

부모가 출력

자식들이 출력

Assignment 3-1 (cont'd)

결과화면 (cont'd)

```
===== Disconnected client =====
[Wed May 15 18:10:23 2019]
IP : 223.195.33.43
Port : 22343
=====
===== Connection History =====
NO.      IP          PID      PORT      TIME
1        222.222.23.55  2335     22333     Wed May 15 18:03:20 2019
[Wed May 15 18:12:20 2019] 2336 process is terminated.
[Wed May 15 18:12:20 2019] 2335 process is terminated.
[Wed May 15 18:12:20 2019] 2334 process is terminated.
[Wed May 15 18:12:20 2019] 2333 process is terminated.
[Wed May 15 18:12:20 2019] 2332 process is terminated.
[Wed May 15 18:12:20 2019] Server is terminated.
```

client 종료 시

History 제목 → 부모 프로세스가 출력
History 내용 → 자식 프로세스가 출력

SIGINT signal 발생

child process 종료 시

server 종료 시

server 종료 시,
client가 먼저 종료되어야 함!

Assignment 3-1 (cont'd)

- **Code Requirements**

- 이전 과제 부분에 문제가 있는 경우 감점
- 출력 형식에 맞지 않으면 감점
- Copy 적발 시 0점

- **Makefile Requirements**

- 실행 파일이 “**preforked_server**”로 생성되도록 Makefile 작성
- 컴파일 도중 warning 발생 시 감점
- “\$ make” 를 통해 실행파일이 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 → assignment3-1
 - **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_preforked_server.c
 - Makefile
 - 실행 파일명 : preforked_server
 - 실행 파일명 다르게 작성 시 감점

- 보고서 및 압축 파일 명 양식


- Web3_1_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_1_A_20231234567.pdf
 - 압축 파일 명: Web3_1_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

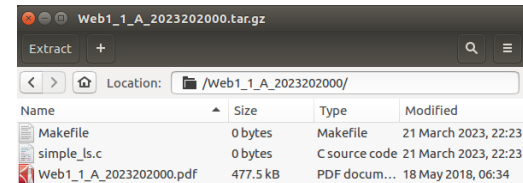
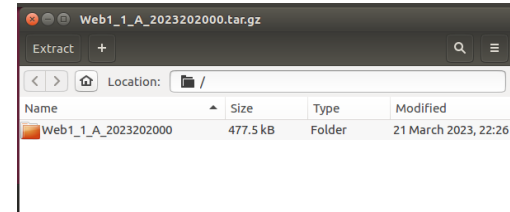
0KB

- 이론 과목에 .txt 파일 미 제출 시 감점
- 과제 제출
 - KLAS – 강의 과제 제출
 - 2023년 5월 17일 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`

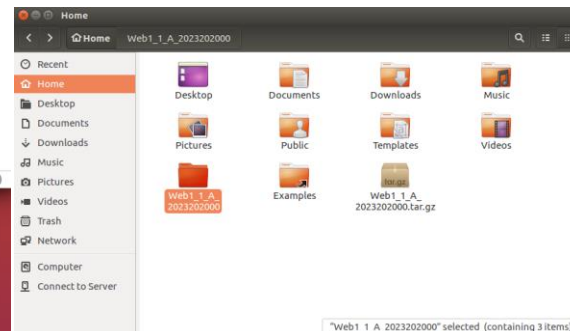
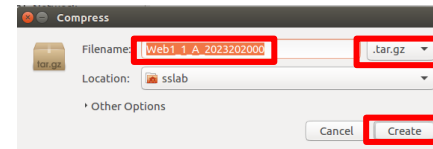
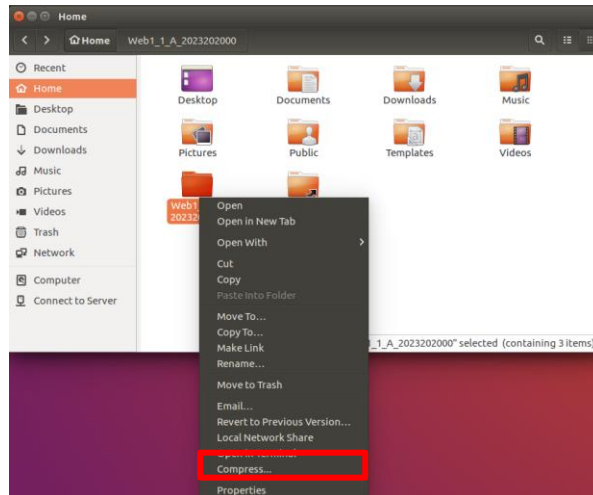
과제 제출 양식

- 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
```



- 디렉토리에서 직접 압축



2023년 1학기 시스템프로그래밍실습

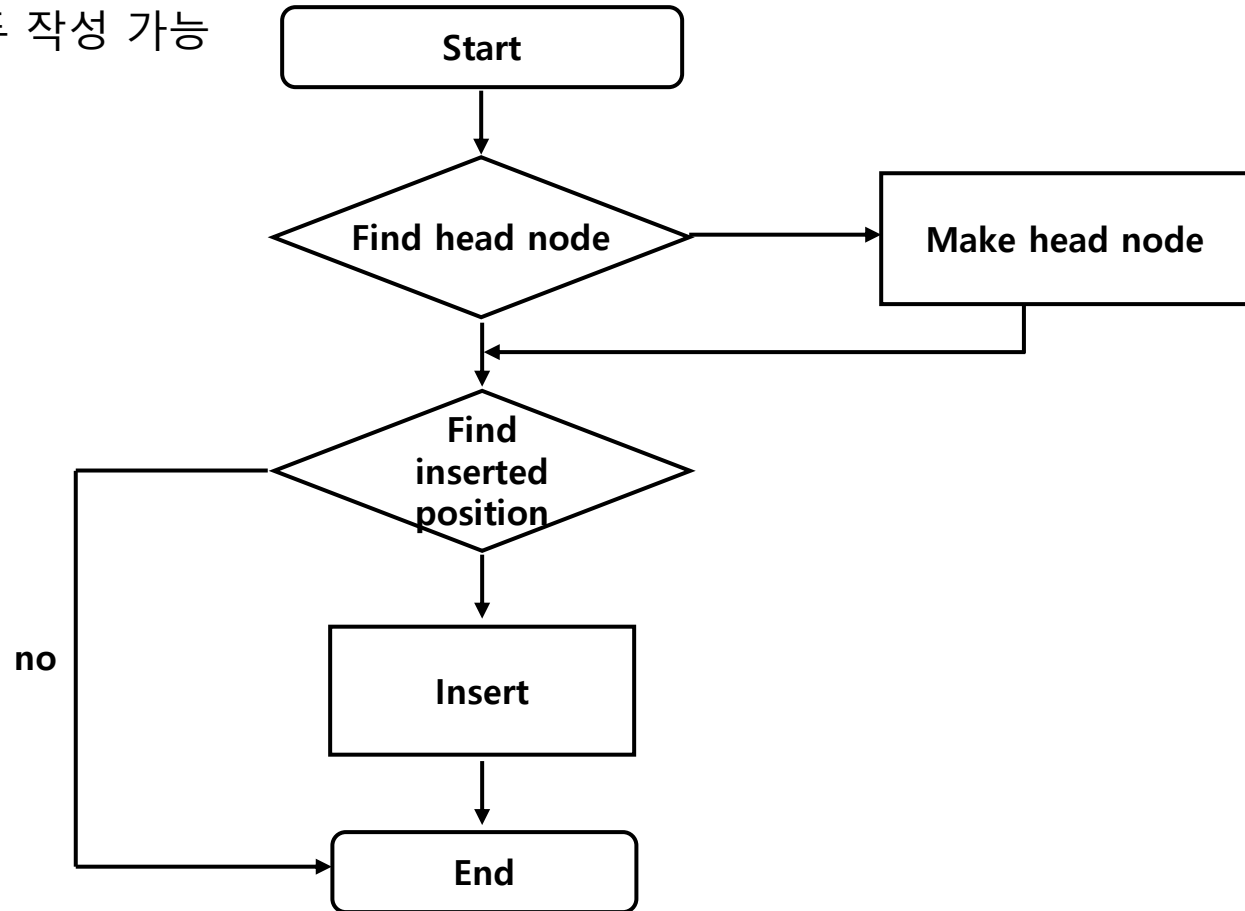
Appendix

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;
```