

2023년 1학기 시스템프로그래밍 & 시스템 프로그래밍 실습

Assignment 3-2

System Software Laboratory
College of Software and Convergence
Kwangwoon Univ.

Assignment 3-2

- Assignment 3-1에 다음의 사항을 변경 및 추가

- httpd.conf에 따라 프로세스, history개수 등을 조절

이름	내용
Maxchilds	최대 자식 프로세스 개수
MaxIdleNum	최대 Idle 프로세스 개수
MinIdleNum	최소 Idle 프로세스 개수
StartProcess	프로그램 시작 시 생성될 프로세스 개수
MaxHistory	출력되는 history의 수

- 10초 마다, 공유메모리에 저장된 history 기록을 최신순서대로 출력
 - 자식 프로세스는 client의 접속이 들어올 때 마다, History정보를 공유메모리에 저장
 - 공유메모리에 저장된 history의 수가 MaxHistory수와 같으면 가장 오래된 history를 제거 후 저장
- 본 프로젝트에서는 다음과 같은 고정된 conf를 사용

```
[httpd.conf]
MaxChilds: 10
MaxIdleNum: 6
MinIdleNum: 4
StartProcess: 5
MaxHistory: 10
```

Assignment 3-2 (cont'd)

- Assignment 3-1에 다음의 사항을 변경 및 추가
 - Process에서 공유메모리에 접근할 때 thread를 생성한 뒤 이를 활용
 - fork를 사용하여 (StartProcess)개의 자식 프로세스 생성 (Assignment 3-1의 방식 유지)
 - Shared memory key value는 40000으로 고정
 - 공유메모리 동기화 문제를 pthread_mutex_lock, pthread_mutex_unlock로 해결
 - Idle process 관리
 - Idle process: client와 연결이 되지 않은 자식 프로세스
 - Idle process 수를 공유메모리로 관리
 - Idle process의 수가 변경되는 경우 부모 프로세스에서 출력
 - Idle process의 수가 (MinIdleNum)개 미만인 되거나 (MaxIdleNum)개 초과가 되면 process를 생성 또는 종료하여 5개를 유지
 - Client와 연결된 자식 프로세스의 수와 idle process의 수의 합이 'Maxchilds'를 넘지 않도록 조절

Assignment 3-2 (cont'd)

- **Assignment 3-1에 다음의 사항을 변경 및 추가 (cont'd)**
 - 프로그램 종료 전 shared memory 제거 확인(SIGINT signal에서도 처리)
 - Client가 연결을 종료하면 자식 프로세스에서 종료 전 5초 sleep (by sleep() function)
 - 기타 내용은 결과 예제 참고

Assignment 3-2 (cont'd)

- Server start
- read httpd.conf

[httpd.conf]
MaxChilds: 10
MaxIdleNum: 6
MinIdleNum: 4
StartProcess: 5
MaxHistory: 10

Child
process 생성

Client 접근

idle < 4 or
idle > 6

yes

Process 생성 또는 종료를 통해 idle
process 수를 5개로 변경 및 출력

no

Child
상태 변경

no

Server
stop

yes

Idle process 수
변경 및 출력

Child process 및
idle process 종료
공유메모리 제거

Assignment 3-2 (cont'd)

결과 화면

```
[Wed May 11 17:49:03 2019] Server is started.

[Wed May 11 17:49:03 2019] 2134 process is forked.
[Wed May 11 17:49:03 2019] IdleProcessCount : 1
[Wed May 11 17:49:03 2019] 2135 process is forked.
[Wed May 11 17:49:03 2019] IdleProcessCount : 2
[Wed May 11 17:49:03 2019] 2136 process is forked.
[Wed May 11 17:49:03 2019] IdleProcessCount : 3
[Wed May 11 17:49:03 2019] 2137 process is forked.
[Wed May 11 17:49:03 2019] IdleProcessCount : 4
[Wed May 11 17:49:03 2019] 2138 process is forked.
[Wed May 11 17:49:03 2019] IdleProcessCount : 5
...
===== New client =====
[Wed May 11 17:52:08 2019]
IP : 128.134.123.45
Port : 22345
=====

[Wed May 11 17:52:08 2019] IdleProcessCount : 4
...
```

StartProcess 값 만큼 프로세스 생성

Assignment 3-2 (cont'd)

결과 화면 (cont'd)

```
===== New client =====
```

```
[Wed May 11 17:52:12 2019]
```

```
IP : 128.134.123.43
```

```
Port : 22343
```

MinIdleNum > IdleProcessCount일 경우

```
[Wed May 11 17:52:12 2019] IdleProcessCount : 3
```

```
[Wed May 11 17:52:12 2019] 2139 process is forked.
```

```
[Wed May 11 17:52:12 2019] IdleProcessCount : 4
```

```
[Wed May 11 17:52:12 2019] 2140 process is forked.
```

```
[Wed May 11 17:52:12 2019] IdleProcessCount : 5
```

```
...
```

Assignment 3-2 (cont'd)

결과 화면 (cont'd)

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

```
[Wed May 11 17:52:17 2019]
```

```
IP : 128.134.123.45
```

```
Port : 22345
```

```
[Wed May 11 17:52:17 2019] IdleProcessCount : 6
```

```
...
```

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

```
[Wed May 11 17:52:19 2019]
```

```
IP : 128.134.123.43
```

```
Port : 22343
```

MaxIdleNum < IdleProcessCount일 경우

```
[Wed May 11 17:52:19 2019] IdleProcessCount : 7
```

```
[Wed May 11 17:52:19 2019] 2134 process is terminated.
```

```
[Wed May 11 17:52:19 2019] IdleProcessCount : 6
```

```
[Wed May 11 17:52:19 2019] 2135 process is terminated.
```

```
[Wed May 11 17:52:19 2019] IdleProcessCount : 5
```

```
...
```

```
===== New client =====
```

```
[Wed May 11 17:52:12 2019]
```

```
IP : 128.134.123.36
```

```
Port : 22347
```


Assignment 3-2 (cont'd)

결과 화면 (cont'd)

이전 과제와 다르게 History를 공유메모리에 저장하므로, 'NO.'가 순차적으로 작성되었음.

```
...
===== Connection History =====
NO.      IP          PID      PORT      TIME
1        223.195.33.36   2336     22347     Wed May 11 18:07:34 2019
2        223.195.33.43   2335     22343     Wed May 11 18:05:34 2019
3        223.195.33.45   2334     22345     Wed May 11 18:03:20 2019
=====
...
[Wed May 11 18:53:02 2019] 2140 process is terminated.
[Wed May 11 18:53:02 2019] IdleProcessCount : 4
[Wed May 11 18:53:02 2019] 2139 process is terminated.
[Wed May 11 18:53:02 2019] IdleProcessCount : 3
[Wed May 11 18:53:02 2019] 2138 process is terminated.
[Wed May 11 18:53:02 2019] IdleProcessCount : 2
[Wed May 11 18:53:02 2019] 2137 process is terminated.
[Wed May 11 18:53:02 2019] IdleProcessCount : 1
[Wed May 11 18:53:02 2019] 2136 process is terminated.
[Wed May 11 18:53:02 2019] IdleProcessCount : 0
[Wed May 11 18:53:02 2019] Server is terminated.
```

Assignment 3-2 (cont'd)

- **Code Requirements**

- 이전 과제 부분에 문제가 있는 경우 감점
- 출력 형식에 맞지 않으면 감점

- **Makefile Requirements**

- 실행 파일이 “**ipc_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 → assignment3-2
 - **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_ipc_server.c
 - Makefile
 - 실행 파일명 : ipc_server
 - 실행 파일명 다르게 작성 시 감점

- 보고서 및 압축 파일 명 양식


- Web3_2_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_2_A_20231234567.pdf
 - 압축 파일 명: Web3_2_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

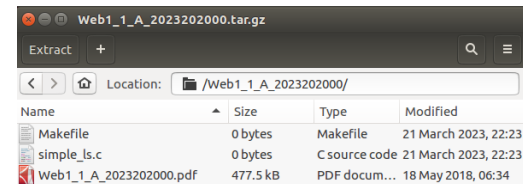
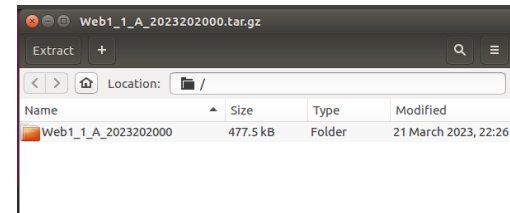
OKB

- 이론 과목에 .txt 파일 미 제출 시 감점
- 과제 제출
 - KLAS – 강의 과제 제출
 - 2023년 5월 24일 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`

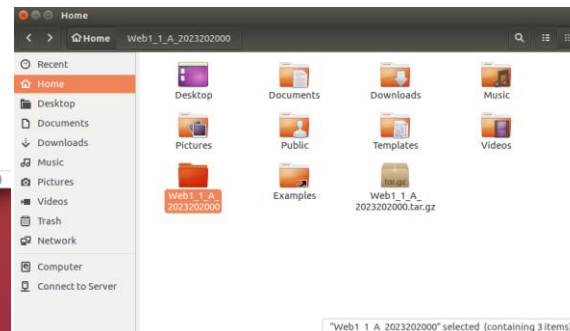
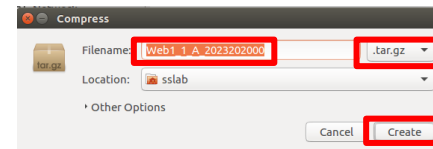
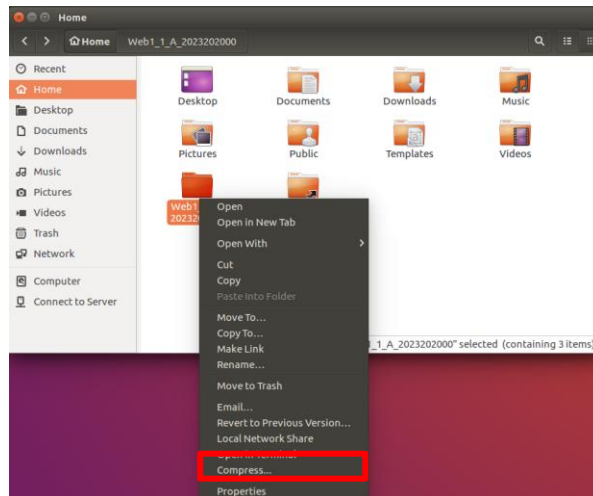
과제 제출 양식

- 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

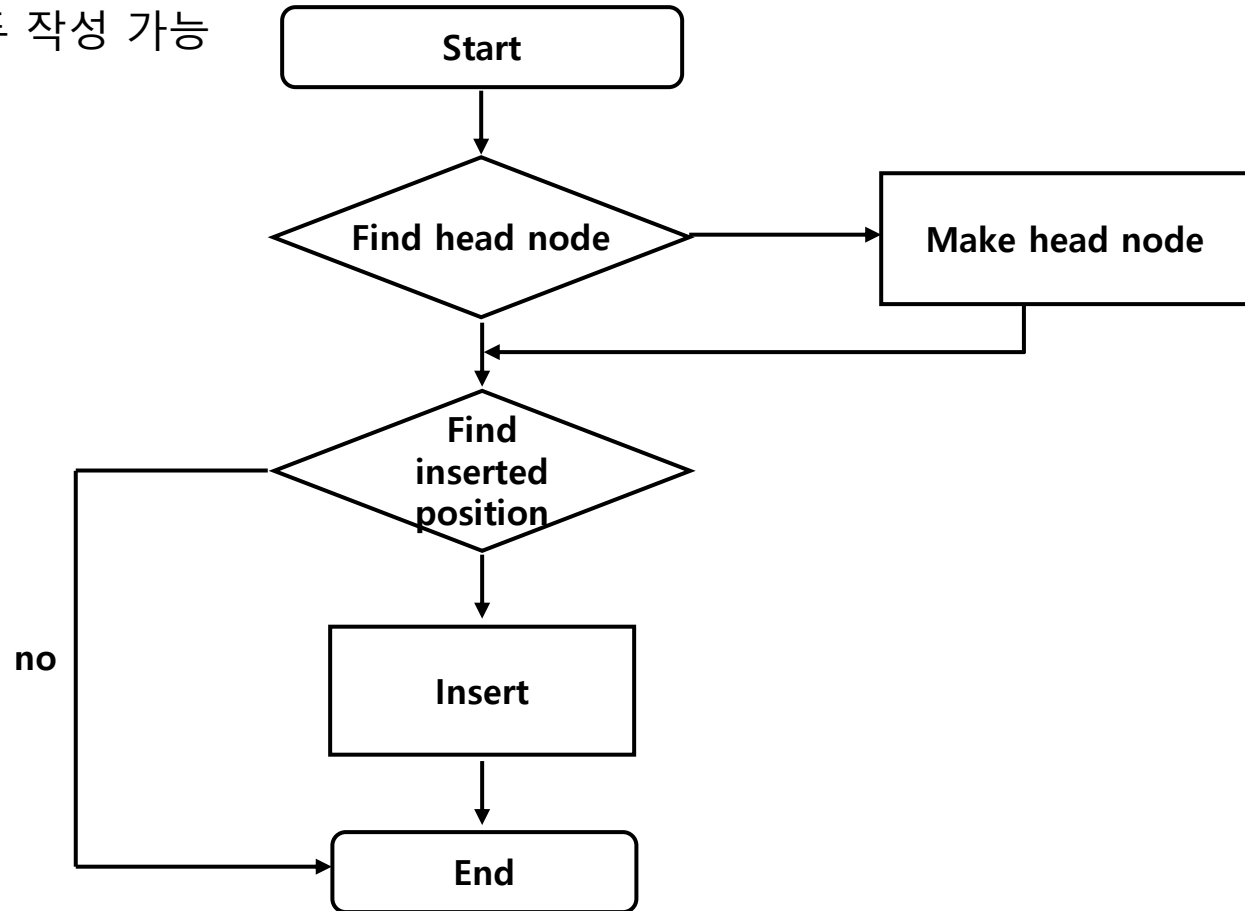
Appendix

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;
```