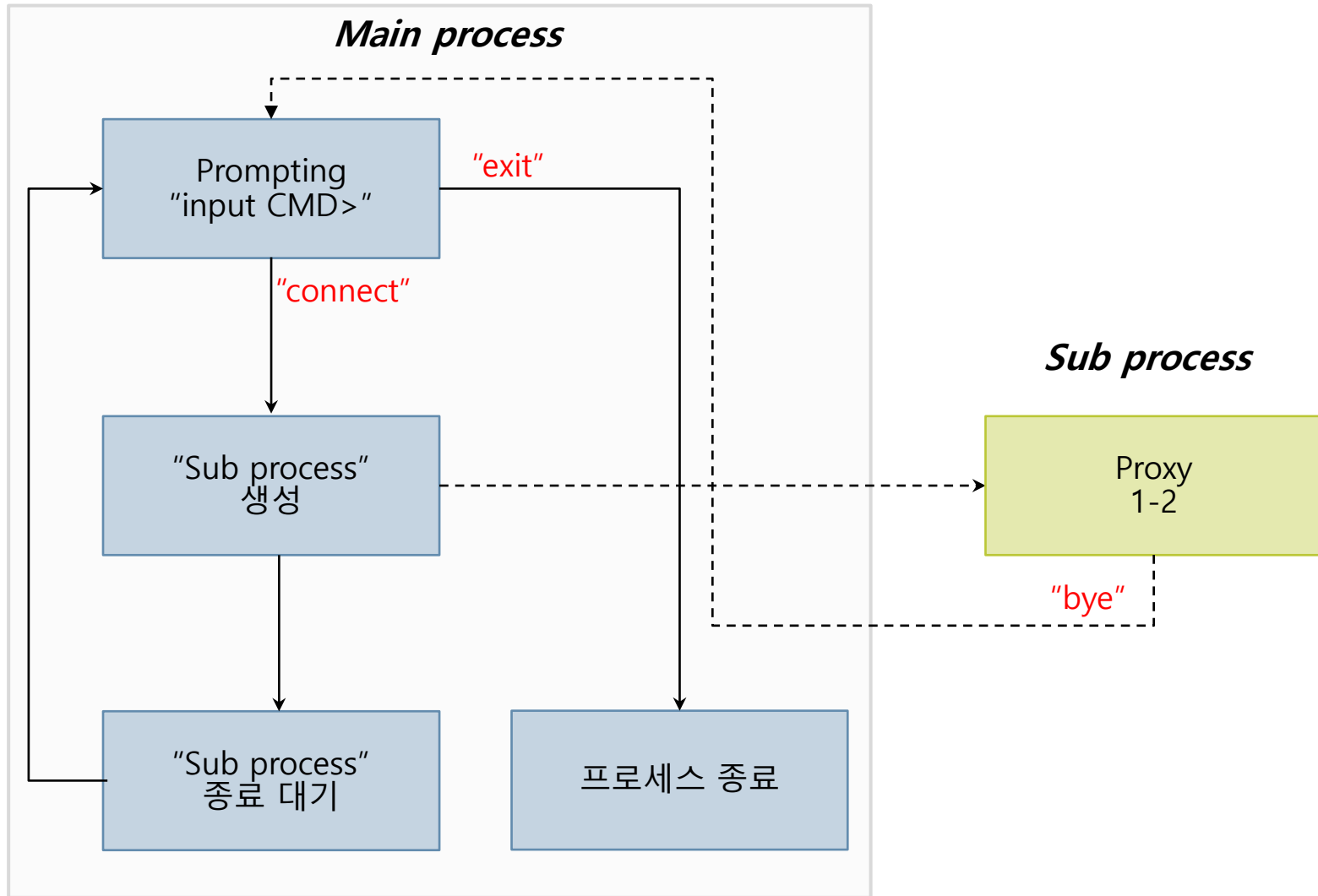


2022년 1학기 시스템프로그래밍

Proxy #1-3

System Software Laboratory
College of Software and Convergence
Kwangwoon Univ.

Proxy 1-3(1/4)



Proxy 1-3(2/4)

- **Concurrent Server Implementation with `fork()`**
- **Main process**
 - 사용자 요청 처리를 위한 새로운 프로세스("Sub process")를 생성하고, 관리
 - 동작:
 - 터미널에 `[(pid)]input CMD>`를 출력하고, 사용자의 명령어 입력을 대기
 - `[(pid)]` → `getpid()` 통해 얻은 값 사용
 - **`connect` 명령어 입력**
 - 새로운 프로세스를 생성하고, 해당 프로세스의 종료까지 대기
 - **`quit` 명령어 입력 시**
 - 프로세스 종료
 - 동작 시간, 생성한 child 프로세스 수 정보에 대한 log를 logfile에 아래와 같이 출력 해야 함
 - ****SERVER** [Terminated] run time: 20 sec. #sub process: 2**

```
sslab@ubuntu:~$ ./proxy_cache
[3933]input CMD> connect
[3934]input URL> bye
[3933]input CMD> quit
sslab@ubuntu:~$
```

Proxy 1-3(3/4)

- **Concurrent Server Implementation with `fork()`**
- **Sub process**
 - 사용자의 URL을 입력 받고, Proxy 1-2에서의 연산을 수행
 - 동작:
 - **터미널에 `[(pid)]input URL>`를 출력하고, 사용자의 명령어 입력을 대기**
 - **`[(pid)] → getpid()` 통해 얻은 값 사용**
 - **bye 입력**
 - 해당 프로세스 종료
 - **URL 입력**
 - Proxy 1-2에서의 연산 수행
 - SHA-1 function (input_url to hashed_url)
 - Check: HIT or MISS
 - Manipulate cache directory
 - logging
 - **참고 사항**
 - 이전 "Sub process"가 생성한 cache file도 유지
 - Log file은 1개만 유지

```
sslab@ubuntu:~$ ./proxy_cache
[3933]input CMD> connect
[3934]input URL> www.kw.ac.kr
[3934]input URL> www.google.com
[3934]input URL> bye
[3933]input CMD> connect
[3935]input URL> bye
[3933]input CMD> quit
```

Proxy 1-3(4/4)

- Example

```
sslab@ubuntu:~$ ./proxy_cache
```

```
[3933]input CMD> connect
```

```
[3934]input URL> www.kw.ac.kr
```

```
[3934]input URL> www.google.com
```

```
[3934]input URL> bye
```

```
[3933]input CMD> connect
```

```
[3935]input URL> www.kw.ac.kr
```

```
[3935]input URL> www.naver.com
```

```
[3935]input URL> bye
```

```
[3933]input CMD> quit
```

```
sslab@ubuntu:~$ cat ~/logfile/logfile.txt
```

```
[Miss]www.kw.ac.kr-[2022/03/26, 23:25:22]
```

```
[Miss]www.google.com-[2022/03/26, 23:25:25]
```

```
[Terminated] run time: 11 sec. #request hit : 0, miss : 2
```

```
[Hit]e00/0f293fe62e97369e4b716bb3e78fababf8f90-[2022/03/26, 23:25:30]
```

```
[Hit]www.kw.ac.kr
```

```
[Miss]www.naver.com-[2022/03/26, 23:25:33]
```

```
[Terminated] run time: 7 sec. #request hit : 1, miss : 1
```

```
**SERVER** [Terminated] run time: 20 sec. #sub process: 2
```

```
sslab@ubuntu:~$
```

Report Requirements

- **Ubuntu 16.04.5 Desktop 64bits 환경에서 채점**
- **보고서 구성**
 - **보고서 표지**
 - 수업 명, 과제 이름, 담당 교수님, 학번, 이름, 강의 시간 필히 명시
 - 과제 이름 → Proxy 1-3
 - 강의 시간 → 월요일, 수요일인 경우 : 월요일로 표기
→ 화요일, 목요일인 경우 : 화요일로 표기
 - 아래의 내용은 보고서에 필히 포함
 - Introduction
 - 과제 소개 – 4줄 이상(background 제외) 작성
 - Flow Chart
 - 코드 작성 순서도
 - Pseudo code
 - 알고리즘
 - 결과화면
 - 수행한 내용을 캡처 및 설명
 - 고찰
 - 과제를 수행하면서 느낀점 작성
 - Reference
 - 과제를 수행하면서 참고한 내용을 구체적으로 기록
 - 강의자료만 이용한 경우 생략 가능

Report Requirements

- **소프트카피만 작성**
 - **제출 파일**
 - **보고서(.pdf 파일) + Source Code(proxy_cache.c) + Makefile**
 - 보고서 이름:
 - **실습번호_학번_수업시간.pdf**
 - 수업시간 월요일, 수요일인 경우 → 1-3_2022722000_mon.pdf
 - 수업시간 화요일, 목요일인 경우 → 1-3_2022722000_tue.pdf
 - C 파일 명:
 - **proxy_cache.c** (Comment 작성)
 - Makefile:
 - 실행파일명: **proxy_cache**
 - **C 파일명, 실행파일명 지정한 이름 외 다른 명으로 작성 시 감점**
- **위 파일들을 압축해서 제출 (파일명:실습번호_학번_수업시간.tar.gz)**
 - 월요일, 수요일 → 1-3_2022722000_mon.tar.gz
 - 화요일, 목요일 → 1-3_2022722000_tue.tar.gz
- **Tar 압축 방법:**
 - 압축 시 : tar -zcvf [압축 파일명].tar.gz [폴더 명]
 - 해제 시 : tar -zxvf 파일명.tar.gz
- 컴파일은 무조건 **Makefile(makefile)**을 이용한 **make**로 함.
 - **Makefile(makefile) 없거나 실행 불가시 0점**
 - **파일 압축 오류 시, 0점 처리**

Report Requirements

- 과제 제출

- On-line 제출
 - KLAS - 강의 과제 제출
 - 2022년 4월 13일 23:59:59 까지
 - 딜레이 받지 않음
 - 제출 마감 시간 내 미제출시 해당과제 0점 처리
 - 교내 서버 문제 발생시, 메일로 과제 제출 허용

- 수업시간 외 과제 질문

- 수업시간 외 과제 질문은 “강의 묻고 답하기” 게시판을 통해 진행
- 과제 제출 마감날 전날까지 업로드 된 질문에만 답변

- 실습수업을 수강하는 학생이 이론 과목에 과제 제출 시

- 간단한 txt 파일로 제출

Ex.) 실습수업 때 과제 제출했습니다.

- 이론 과목에 간단한 txt 파일 미 제출 시 감점