

2025년 1학기 시스템프로그래밍실습 5주차

Get local time & Log file

System Software Laboratory
College of Software and Convergence
Kwangwoon Univ.

과제 세부 일정

■ 1st Assignments

- ~~Overview & Create Cache Directory~~
- **Get local time & log file**
- Create server process

■ 2nd Assignments

- Implement server/client (socket programming)
- Forward HTTP request to Web server & print the HTTP response
- Integrate server side and client side into proxy server

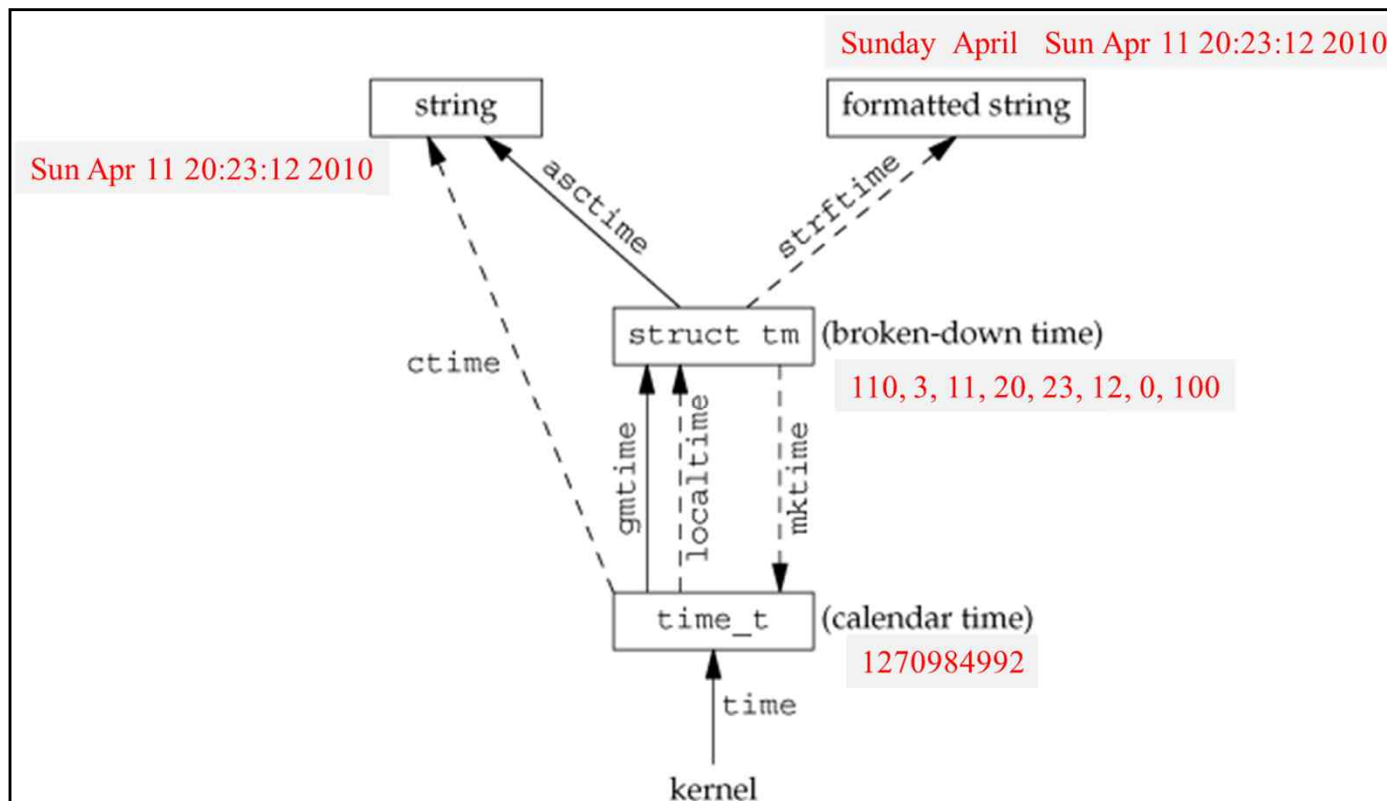
■ 3rd Assignments

- Add cache(assignment_1) to proxy server(assignment_2)
- Synchronize the shared resource

Time and Data (1/9)

- **Data Type of Time**

- `time_t`
 - “time.h”에 정의
 - 유닉스나 POSIX방식의 OS에서 32, 또는 64bit 정수 혹은 실수형으로 구현
 - `time()` 함수를 통해 초단위의 데이터를 받아오기 위한 자료형



Time and Data (2/9)

Function for time translation

- Header file; <time.h>

함수	설명	반환 값
char* ctime (const time_t*)	초 단위 시간 정보를 문자열로 변환	<ul style="list-style-type: none">- 성공 시 각각 해당 데이터 형의 시간 정보 변환- 실패 시 NULL (mktime()은 -1)값을 반환
struct tm* gmtime (const time_t*)	<ul style="list-style-type: none">- 초 단위 시간 정보를 struct tm 형으로 변환- UTC를 기준으로 시간을 할당	
struct tm* localtime (const time_t*)	<ul style="list-style-type: none">- 초 단위 시간 정보를 struct tm 형으로 변환- 시스템 로컬 시간을 기준으로 시간을 할당 (e.g. GMT+09:00 in seoul)	
char* asctime (const struct time_t*)	struct tm 시간 정보를 문자열로 변환	
time_t mktime (struct tm*)	tm 시간 정보를 초 단위시간 정보로 변환	

Time and Data (3/9)

- **time()**

```
#include <time.h>

time_t time(const time_t *timer);
```

- Description
 - The basic time service provided by Unix kernel
 - Get the current calendar time as a value of type `time_t`.
 - The value returned generally represents the number of seconds since 00:00 hours, Jan 1, 1970 UTC (i.e., the current *unix timestamp*).
- Returns value
 - time value if OK, -1 on error

Time and Data (4/9)

- **Year 2038 Problem (i.e. Unix Millennium Bug)**
 - “Unix systems will interpret as a time on 13 December 1901 rather than 19 January 2038.”
 - Reasons – “Integer overflow”
 - *time_t* stores only signed 32-bit integer.
 - So, implementations of *time_t* **cannot encode times after 03:14:07 UTC on 19 January 2038**
 - i.e. 2,147,483,647 seconds after 1 January 1970
 - Solutions
 - Usage of signed 64-bit *time_t* integer value
 - With this type, we can express time to **15:30:08 on Sunday, 4 December 292,277,026,596.**
 - More detail...
 - http://en.wikipedia.org/wiki/Year_2038_problem

Time and Data (5/9)

- **ctime()**

```
#include <time.h>
```

```
char *ctime(const time_t *timer);
```

- Description
 - Produces the familiar string that represents the time
 - Tue Jan 14 17:49:03 1992\n\0
- Returns a pointer to null terminated string

Time and Data (6/9)

- **gmtime()**

```
#include <time.h>

struct tm *gmtime(const time_t *timer);
```

- Description

- Converts a calendar time into a broken-down time (tm structure).
- Uses the value pointed by timer to fill a tm structure with the values that represent the corresponding time, expressed as a UTC time (i.e., the time at the GMT timezone).

- Return value

- On success, a pointer to a tm structure
- On error, NULL

Time and Data (7/9)

- **localtime()**

```
#include <time.h>

struct tm *localtime(const time_t *timer);
```

- Description
 - It converts time_t to struct tm with current time zone and daylight.
 - Uses the value pointed by timer to fill a tm structure with the values that represent the corresponding time, expressed for the local timezone
- Return value
 - On success, a pointer to a tm structure
 - On error, NULL

Time and Data (8/9)

- **struct tm**

```
struct tm          /* a broken-down time */
{
    int tm_sec;      /* 초 [0,59] */
    int tm_min;      /* 분 [0,59] */
    int tm_hour;     /* 시 [0,23] */
    int tm_mday;     /* 일 [1,31] */
    int tm_mon;      /* 월 [0,11] */
    int tm_year;     /* 년 since 1900 */
    int tm_wday;     /* 요일; 일요일부터 [0,6] */
    int tm_yday;     /* 날짜 [0,365] */
    int tm_isdst;    /* Summer time flag: <0, 0, >0 */
};
```

(실습 1)Time (9/9)

- E.g.

```
1 #include <stdio.h>
2 #include <time.h>
3
4 int main(void) {
5     time_t now;
6     struct tm *ltp, *gtp;
7
8     time(&now);
9     ltp = localtime(&now);
10    printf("ctime(local):      %s\n", ctime(&now));
11    printf("localtime(local) : %02d:%02d:%02d\n", ltp->tm_hour, ltp->tm_min, ltp->tm_sec);
12    gtp = gmtime(&now);
13    printf("asctime(GMT)       : %s\n", asctime(gtp));
14    printf("gmtime (GMT)       : %02d:%02d:%02d\n", gtp->tm_hour, gtp->tm_min, gtp->tm_sec);
15    printf("mktime           : %d\n", mktime(gtp));
16
17    return 0;
18 }
```

```
ctime(local)      : Wed Mar 21 04:05:11 2018
localtime(local)  : 04:05:11
asctime(GMT)      : Wed Mar 21 11:05:11 2018
gmtime (GMT)      : 11:05:11
mktime           : 1521659111
```

2025년 1학기 시스템프로그래밍 & 시스템프로그래밍실습

Proxy #1-2

System Software Laboratory
College of Software and Convergence
Kwangwoon Univ.

proxy 1-2 (1/5)

- 구현된 Proxy 1-1에 Proxy 1-2를 구현

- **Description**

- **Create a Directory and Log text File**

- 디렉토리 logfile 경로 : ~/logfile
 - logfile.txt 경로 : ~/logfile/logfile.txt
 - Time : time when receiving URL (localtime() 사용)
 - logfile.txt format

- Hit일 경우

- [Hit]Directory name/file name-[Time] (Time은 year/month/day, hour:min:sec 으로 표기)

- [Hit]URL (URL은 입력한 URL)

- e.g.

- [Hit]e00/0f293fe62e97369e4b716bb3e78fababf8f90-[2025/1/1, 10:26:12]

- [Hit]www.kw.ac.kr

- Miss일 경우

- [Miss]URL-[Time]

- e.g. [Miss]ce.kw.ac.kr-[2025/1/1, 11:37:14]

- 프로그램 실행 시간 , Request 횟수를 프로그램이 종료될 때 기록

- Time()를 사용하여 프로그램 실행시간을 기록
 - Request 횟수 : 프로그램이 종료 될 때 까지의 Hit과 miss의 횟수 기록
 - Format :

[Terminated] run time: 7 sec. #request hit : 2, miss : 3

proxy 1-2 (1/5)

- 구현된 Proxy 1-1에 Proxy 1-2를 구현
- **Description**
 - **HIT / MISS 판별**
 - Cache file의 path를 opendir() 과 readdir() 을 사용하여 구함
 - cache directory 내의 모든 파일의 파일명과 hashed URL이 동일한지 비교
 - 같을 때 : HIT
 - 다를 때 : MISS

Proxy 1-2 (3/5)

- Input

```
$ ./proxy_cache
input URL> www.kw.ac.kr
input URL> www.naver.com
input URL> www.google.com
input URL> www.kw.ac.kr
input URL> www.naver.com
input URL> klas.kw.ac.kr
input URL> bye
$
```

Proxy 1-2 (4/5)

Cache Directory

```
$ ls -R ~/cache
/home/sslslab/cache/3ef:
9fd210fb8e00c8114ff978d282258ed8a48ea

/home/sslslab/cache/d8b:
99f68b208b5453b391cb0c6c3d6a9824f3c3a

/home/sslslab/cache/e00:
0f293fe62e97369e4b716bb3e78fababf8f90

/home/sslslab/cache/fed:
818da7395e30442b1dcf45c9b6669d1c0ff6b
```

```
$ tree ~/cache
/home/sslslab/proxy_cache/
├── 3ef
│   └── 9fd210fb8e00c8114ff978d282258ed8a48ea
├── d8b
│   └── 99f68b208b5453b391cb0c6c3d6a9824f3c3a
├── e00
│   └── 0f293fe62e97369e4b716bb3e78fababf8f90
└── fed
    └── 818da7395e30442b1dcf45c9b6669d1c0ff6b

4 directories, 4 files
```


Proxy 1-2 (5/5)

- Log file

```
$ cat ~/logfile/logfile.txt
[Miss]www.kw.ac.kr-[2025/02/23, 18:14:29]
[Miss]www.naver.com-[2025/02/23, 18:14:31]
[Miss]www.google.com-[2025/02/23, 18:14:33]
[Hit]e00/0f293fe62e97369e4b716bb3e78fababf8f90-[2025/02/23, 18:14:35]
[Hit]www.kw.ac.kr
[Hit]fed/818da7395e30442b1dcf45c9b6669d1c0ff6b-[2025/02/23, 18:14:38]
[Hit]www.naver.com
[Miss]ce.kw.ac.kr-[2025/02/23, 18:14:40]
[Terminated] run time: 14 sec. #request hit : 2, miss : 4
$
```

Report Requirements

- **Ubuntu 20.04.6 Desktop 64bits 환경에서 채점**
- **Copy 발견 시 0점 처리**
- **보고서 구성**
 - **보고서 표지**
 - 수업 명, 과제 이름, 담당 교수님, 학번, 이름, 강의 시간 필히 명시
 - 과제 이름 → Proxy 1-2
 - **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 느낀 점 작성
 - 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.xz)
 - 1)보고서:
 - 보고서를 **pdf로 변환**하여 제출
 - 보고서 이름은 *Proxy1-2_수강분류코드_학번_이름* 으로 작성
 - 2)C 파일 명:
 - **proxy_cache.c**
 - **Comment 작성(Appendix 내용 참고)**
 - 3)Makefile:
 - 실행파일명: **proxy_cache**
 - C 파일명, 실행파일명 지정한 이름 외 다른 명으로 작성 시 감점


- **tar.xz 압축 방법**

- **(Appendix 내용 참고)**

- 컴파일은 무조건 Makefile(makefile)을 이용한 make로 함.
 - Makefile(makefile) 없거나 실행 불가시 0점
 - 파일 압축 오류 시, 0점 처리

Report Requirements

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

 실습수업때 제출했습니다.

2022-08-29 오후 3:58

텍스트 문서

0KB

- 이론 과목에 .txt 파일 미 제출 시 감점
- .tar.xz 파일로 제출 하지 않을 시 감점

- 예시-이론 월5 수6 수강하는 학생인 경우
 - 보고서: Proxy1-2_A_2025123456_홍길동.pdf
 - 압축 파일 명: Proxy1-2_A_2025123456_홍길동.tar.xz

수강요일	이론1 월5수6	이론2 목4	실습1 목12
수강분류 코드	A	B	C

- 과제 제출
 - KLAS – 강의 과제 제출**
 - 2025년 4월 10일 목요일 23:59까지 제출**
 - 딜레이 받지 않음
 - 제출 마감 시간 내 미제출시 해당 과제 **0점 처리**
 - 교내 서버 문제 발생 시, 메일로 과제 제출 허용