KWANGWOON UNIVERSITY

2022년 1학기 **시스템프로그래밍실습** 5주차

Get local time & Log file

System Software Laboratory
College of Software and Convergence
Kwangwoon Univ.

1st Assignment's Descriptions

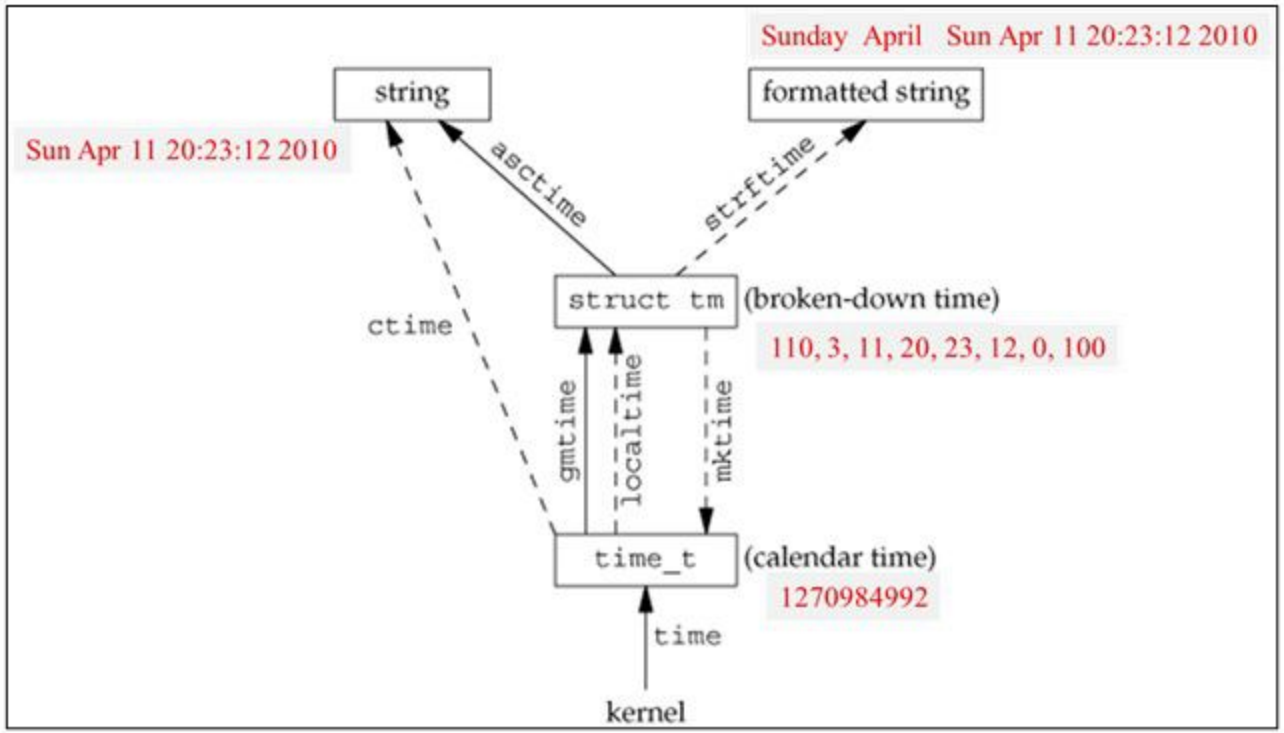
- **Assignment 1-1**
  - 표준입력(STDIN)으로부터 URL 입력
  - SHA-1 Algorithm을 사용하여 textual URL을 Hashed URL로 변환
  - Hashed URL을 이용하여 Directory와 File 생성

- **Assignment 1-2**
  - 시스템으로부터 현재 시간 구함
  - Log file을 생성
  - Log file에 입력 URL과 현재 시간 기록

- **Assignment 1-3**
  - Multiple Processing

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 *time)	- 초 단위 시간 정보를 문자열로 변환	- 성공 시 각각 해당 데이터형의 시간 정보 변환 - 실패 시 NULL (mktime()은 -1)값을 반환
struct tm* **gmtime**(const time_t *time)	- 초 단위 시간 정보를 struct tm 형으로 변환 - UTC를 기준으로 시간을 할당	
struct tm* **localtime**(const time_t *timer)	- 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 tm *time)	- struct tm 시간 정보를 문자열로 변환	
time_t **mktime**(struct tm *time)	- 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.

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

int main(void) {
    time_t now;
    struct tm *ltp, *gtp;

    time(&now);
    ltp = localtime(&now);
    printf("ctime(local):      %s\n", ctime(&now));
    printf("localtime(local) : %02d:%02d:%02d\n", ltp->tm_hour, ltp->tm_min, ltp->tm_sec);
    gtp = gmtime(&now);
    printf("asctime(GMT)     : %s\n", asctime(gtp));
    printf("gmtime (GMT)     : %02d:%02d:%02d\n", gtp->tm_hour, gtp->tm_min, gtp->tm_sec);
    printf("mktime           : %d\n", mktime(gtp));

    return 0;
}
```

```
ctime(local)      :  Thu Mar 24 20:08:45 2022

localtime(local) :  20:08:45
asctime(GMT)     :  Fri Mar 25 03:08:45 2022

gmtime (GMT)     :  03:08:45
mktime           :  1648206525
```

KWANGWOON UNIVERSITY

2022년 1학기 **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-[2022/1/1, 10:26:12]
        [Hit]www.kw.ac.kr

      - **Miss일 경우**

        [Miss]URL-[Time]
        e.g. [Miss]ce.kw.ac.kr-[2021/1/1, 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/sslab/cache/3ef:
9fd210fb8e00c8114ff978d282258ed8a48ea

/home/sslab/cache/d8b:
99f68b208b5453b391cb0c6c3d6a9824f3c3a

/home/sslab/cache/e00:
0f293fe62e97369e4b716bb3e78fababf8f90

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

```
$ tree ~/cache
/home/sslab/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-[2022/02/23, 18:14:29]
[Miss]www.naver.com-[2022/02/23, 18:14:31]
[Miss]www.google.com-[2022/02/23, 18:14:33]
[Hit]e00/0f293fe62e97369e4b716bb3e78fababf8f90-[2022/02/23, 18:14:35]
[Hit]www.kw.ac.kr
[Hit]fed/818da7395e30442b1dcf45c9b6669d1c0ff6b-[2022/02/23, 18:14:38]
[Hit]www.naver.com
[Miss]klas.kw.ac.kr-[2022/02/23, 18:14:40]
[Terminated] run time: 14 sec. #request hit : 2, miss : 4
$
```