Miscellaneous

경희대학교 컴퓨터공학과

조 진 성

Variable Argument Lists

- Include <stdarg.h> file
  - ✓ `#include <stdarg.h>`
- Declaration of a variable of type `va_list`
  - ✓ `va_list arglist;`
- Start to get variable arguments
  - ✓ `va_start(arglist, lastfix);`
  - ✓ The second parameter, lastfix
    - Last fixed argument
- Get variable arguments
  - ✓ `arg = va_arg(arglist, type);`
  - ✓ The second parameter, type
    - You have to know it somehow
    - e.g.) predefined or from format string in printf
- Stop to get variable arguments
  - ✓ `va_end(arglist)`

Exercise

- Variable argument examples

```
$ gcc -o varg varg.c (or make varg)
$ ./varg
```

Time and Date

- Count the number of seconds that have passed since the Epoch:
  - ✓ 00:00:00 January 1, 1970, UTC (Coordinated Universal Time)

- Data structure for time and date

```
#include <time.h>
struct tm  {
   int tm_sec;
   int tm_min;
   int tm_hour;
   int tm_mday;
   int tm_mon;
   int tm_year;
   int tm_wday;   /* days since Sunday: [0, 6) */
   int tm_yday;   /* days since January 1: [0. 365) */
   int tm_isdst;  /* daylight saving time flag: <0, 0, >0 */
}
```

Time and Date (Cont'd)

- **Functions**
  - ✓ `#include <time.h>`
  - ✓ `time_t time(time_t *calptr);`
  - ✓ return: value of time if OK, -1 on error

  - ✓ `struct tm *gmtime(time_t *calptr);`
  - ✓ `struct tm *localtime(time_t *calptr);`
  - ✓ both return: pointer to broken-down time
  - ✓ localtime takes account the local time zone and daylight saving time flag

  - ✓ `time_t mktime(struct tm *tmptr);`
  - ✓ return: calendar time if OK, -1 on error

  - ✓ `char *asctime(struct tm *tmptr);`
  - ✓ `char *ctime(time_t *calptr);`
  - ✓ return: pointer to null terminated string

Time and Date (Cont'd)

- **Functions (Cont'd)**
  - ✓ `#include <time.h>`
  - ✓ `size_t strftime(char *buf, size_t maxsize, char *format, struct tm *tmptr);`
  - ✓ return: number of characters stored in array if OK, else 0

  - ✓ Conversion specifiers for strftime
    - `%a` : abbreviated weekday name (Tue)
    - `%A` : full weekday name (Tuesday)
    - `%b` : abbreviated month name (Jan)
    - `%c` : date and time (Tue Jan 14 19:40:30 2003)
    - `%d` : day of the month (14)
    - `%H` : hour of the 24-hour day (19)
    - `%I` : hour of the 12-hour day (07)
    - `%j` : day of the year (014)
    - `%m` : month (01)

Time and Date (Cont'd)

■ Functions (Cont'd)

  ✓ Conversion specifiers for strftime (Cont'd)

   ▪ `%M` : minute (40)

   ▪ `%p` : AM/PM (PM)

   ▪ `%S` : second (30)

   ▪ `%U` : Sunday week number (02)

   ▪ `%w` : weekday[Sunday=0] (2)

   ▪ `%x` : date (01/14/03)

   ▪ `%X` : time (19:40:30)

   ▪ `%y` : year without century (03)

   ▪ `%Y` : year with century (2003)

   ▪ `%Z` : time zone name (KST)

  ✓ Cf) `date` command

Exercise

- Time and Date related functions

    $ gcc -o time time.c (or make time)

    $ ./time

Summary

- System calls for variable argument list
  - ✓ `va_start, va_arg, va_end`

- System calls for time and date
  - ✓ `time`
  - ✓ `gmtime, localtime`
  - ✓ `mktime`
  - ✓ `asctime, ctime`
  - ✓ `strftime`