Copying:

memcpy(void * destination, const void * source, size_t num)

Copy block of memory (function)

memmove(void * destination, const void * source, size_t num)

Move block of memory (function)

strcpy(char * destination, const char * source)

Copy string (function)

strncpy(char * destination, const char * source, size_t num)

Copy characters from string (function)

Concatenation:

strcat(char * destination, const char * source)

Concatenate strings (function)

strncat(char * destination, const char * source, size_t num)

Append characters from string (function)

Comparison:

memcmp(const void * ptr1, const void * ptr2, size_t num)

Compare two blocks of memory (function)

strcmp(const char * str1, const char * str2)

Compare two strings (function)

strcoll(const char * str1, const char * str2)

Compare two strings using locale (function)

strncmp(const char * str1, const char * str2, size_t num)

Compare characters of two strings (function)

strxfrm(const char * str1, const char * str2, size_t num)

Transform string using locale (function)

Searching:

memchr(const void * ptr, int value, size_t num)

Locate character in block of memory (function)

strchr(const char * str, int character)

Locate first occurrence of character in string (function)

strcspn(const char * str1, const char * str2)

Get span until character in string (function)

strpbrk(const char * str1, const char * str2)

Locate characters in string (function)

strrchr(const char * str, int character)

Locate last occurrence of character in string (function)

strspn(const char * str1, const char * str2)

Get span of character set in string (function)

strstr(const char * str1, const char * str2)

Locate substring (function)

strtok(char * str, const char * delimiters)

Split string into tokens (function)

Other:

memset(void * ptr, int value, size_t num)

Fill block of memory (function)

strerror(int errnum)

Get pointer to error message string (function)

strlen(const char * str)

Get string length (function)

Macros

NULL

Null pointer (macro)

Types

size_t

Unsigned integral type (type)

名稱	說明
void *memcpy(void *dest, const void *src, size_t n);	將n位元組長的內容從一個記憶體位址複製到另一個位址；如果兩個位址存在重疊，則最終行為未定義
void *memmove(void *dest, const void *src, size_t n);	將n位元組長的內容從一個記憶體位址複製到另一個位址；與memcpy不同的是它可以正確作用於兩個存在重疊的位址
char *strcpy(char* str1, const char* str2);	將str2拷貝給str1
char *strncpy(char* str1, const char* str2, size_t n);	截取str2的n個字元拷貝給str1
char *strncat(char *dest, const char *src, size_t n);	從src截取n個字元連接在字串dest之後，返回dest字串
int memcmp(const void *s1, const void *s2, size_t n);	對從兩個記憶體位址開始的n個字元進行比較
int strcmp(const char *, const char *);	基於字典順序比較兩個字串
int strcoll(const char *, const char *);	基於當前區域設定的字元順序比較兩個字串
int strncmp(const char *, const char *, size_t n);	基於字典順序比較兩個字串，最多比較n個位元組
size_t strxfrm(char *dest, const char *src, size_t n);	根據當前locale轉換一個字串為strcmp使用的內部格式
void *memchr(const void *s, char c, size_t n);	在從s開始的n個位元組內尋找c第一次出現的位址並返回，若未找到則返回NULL
char *strchr(const char* str, int ch);	從字串str頭開始尋找字元ch首次出現的位置
size_t strcspn(const char *s, const char *strCharSet);	從字串s的起始處開始，尋找第一個出現在strCharSet中的字元，返回其位置索引值。換句話說，返回從字串s的起始位置的完全由不屬於strCharSet中的字元構成的子串的最大長度。strcspn為string complement span的縮寫。不支援多位元組字元集。
char *strpbrk(const char *s, const char *breakset);	在字串s中尋找breakset中任意字元第一次出現的位置的指標值。strpbrk為string pointer break縮寫。通常，breakset是分隔符的集合。不支援多位元組字元集。
char *strrchr(const char* str,int ch);	從字串str尾開始尋找字元ch首次出現的位置
size_t strspn(const char *s, const char *strCharSet);	從字串s的起始處開始，尋找第一個不出現在strCharSet中的字元，返回其位置索引值。換句話說，返回從字串s的起始位置的完全由strCharSet中的字元構成的子串的最大長度。strspn為string span的縮寫。不支援多位元組字元集。
char *strstr(const char *haystack, const char *needle);	在字串haystack中尋找字串needle第一次出現的位置，heystack的長度必須長於needle
char *strtok(char *strToken, const char *strDelimit);	將一個字串strToken依據分界符（delimiter）分隔成一系列字串。此函式非執行緒安全，且不可重入；但MSVC實現時使用了thread-local static variable因而是執行緒安全的但仍然是不可重入，即在單執行緒中不能對兩個源字串交替呼叫該函式來分析token，應當對一個字串分析完成後再處理別的字串。
void *memset(void *, int, size_t);	用某種位元組內容覆寫一段記憶體空間
char *strerror(int);	返回錯誤碼對應的解釋字串，參見errno.h（非執行緒安全函式）
size_t strlen(const char *);	返回一個字串的長度
NULL	表示空指標常數的宏，即表示一個不指向任何有效記憶體單元位址的指標常數。
size_t	無符號整型，被用於sizeof運算子的返回值類型。