C 标准库 - <stdlib.h>

简介

stdlib .h 头文件定义了四个变量类型、一些宏和各种通用工具函数。

`<stdlib.h>` 是 C 标准库中的一个头文件，提供了许多通用工具函数，包括内存分配、进程控制、排序和搜索、以及字符串转换等。

库变量

下面是头文件 stdlib.h 中定义的变量类型：

序号	变量 & 描述
1	**size_t** 这是无符号整数类型，它是 **sizeof** 关键字的结果。
2	**wchar_t** 这是一个宽字符常量大小的整数类型。
3	**div_t** 这是 **div** 函数返回的结构。
4	**ldiv_t** 这是 **ldiv** 函数返回的结构。

库宏

下面是头文件 stdlib.h 中定义的宏：

序号	宏 & 描述
1	**NULL** 这个宏是一个空指针常量的值。
2	**EXIT_FAILURE** 这是 exit 函数失败时要返回的值。
3	**EXIT_SUCCESS** 这是 exit 函数成功时要返回的值。
4	**RAND_MAX** 这个宏是 rand 函数返回的最大值。
5	**MB_CUR_MAX** 这个宏表示在多字节字符集中的最大字符数，不能大于 MB_LEN_MAX。

库函数

下面是头文件 stdlib.h 中定义的函数：

序	函数 & 描述
1	double atof(const char *str) 把参数 *str* 所指向的字符串转换为一个浮点数（类型为 double 型）。
2	int atoi(const char *str) 把参数 *str* 所指向的字符串转换为一个整数（类型为 int 型）。
3	long int atol(const char *str) 把参数 *str* 所指向的字符串转换为一个长整数（类型为 long int 型）。

序	函数 & 描述
4	double strtod(const char *str, char **endptr) 把参数 *str* 所指向的字符串转换为一个浮点数（类型为 double 型）。
5	long int strtol(const char *str, char **endptr, int base) 把参数 *str* 所指向的字符串转换为一个长整数（类型为 long int 型）。
6	unsigned long int strtoul(const char *str, char **endptr, int base) 把参数 *str* 所指向的字符串转换为一个无符号长整数（类型为 unsigned long int 型）。
7	void *calloc(size_t nitems, size_t size) 分配所需的内存空间，并返回一个指向它的指针。
8	void free(void *ptr) 释放之前调用 *calloc*、*malloc* 或 *realloc* 所分配的内存空间。
9	void *malloc(size_t size) 分配所需的内存空间，并返回一个指向它的指针。
10	void *realloc(void *ptr, size_t size) 尝试重新调整之前调用 *malloc* 或 *calloc* 所分配的 ptr 所指向的内存块的大小。
11	void abort(void) 使一个异常程序终止。
12	int atexit(void (*func)(void)) 当程序正常终止时，调用指定的函数 **func**。
13	void exit(int status) 使程序正常终止。
14	char *getenv(const char *name) 搜索 name 所指向的环境字符串，并返回相关的值给字符串。
15	int system(const char *string) 由 string 指定的命令传给要被命令处理器执行的主机环境。
16	void *bsearch(const void *key, const void *base, size_t nitems, size_t size, int (compar)(const void *, const void *)) 执行二分查找。
17	void qsort(void *base, size_t nitems, size_t size, int (compar)(const void , const void)) 数组排序。
18	int abs(int x) 返回 x 的绝对值。
19	div_t div(int numer, int denom) 分子除以分母。
20	long int labs(long int x) 返回 x 的绝对值。
21	ldiv_t ldiv(long int numer, long int denom) 分子除以分母。
22	int rand(void) 返回一个范围在 0 到 *RAND_MAX* 之间的伪随机数。
23	void srand(unsigned int seed) 该函数播种由函数 **rand** 使用的随机数发生器。
24	int mblen(const char *str, size_t n) 返回参数 *str* 所指向的多字节字符的长度。
25	size_t mbstowcs(schar_t *pwcs, const char *str, size_t n) 把参数 *str* 所指向的多字节字符的字符串转换为参数 *pwcs* 所指向的数组。
26	int mbtowc(whcar_t *pwc, const char *str, size_t n) 检查参数 *str* 所指向的多字节字符。
27	size_t wcstombs(char *str, const wchar_t *pwcs, size_t n) 把数组 *pwcs* 中存储的编码转换为多字节字符，并把它们存储在字符串 *str* 中。

序	函数 & 描述
28	int wctomb(char *str, wchar_t wchar) 检查对应于参数 *wchar* 所给出的多字节字符的编码。

实例

以下是使用 `<stdlib.h>` 中一些函数的示例。

动态内存分配:

```
#include <stdio.h>
#include <stdlib.h>

int main() {
  int *array = (int*)malloc(10 * sizeof(int));
  if (array == NULL) {
    perror("Memory allocation failed");
    return 1;
  }

  for (int i = 0; i < 10; i++) {
    array[i] = i;
    printf("%d ", array[i]);
  }
  printf("**\n**");

  free(array);
  return 0;
}
```

伪随机数生成:

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

int main() {
  srand(time(NULL));
  for (int i = 0; i < 10; i++) {
    printf("%d ", rand() % 100);
  }
  printf("**\n**");

  return 0;
}
```

字符串转换:

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

int main() {
  const char *str = "12345";
  int num = atoi(str);
  printf("The integer value is %d**\n**", num);

  return 0;
}
```

快速排序:

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

int compare(const void *a, const void *b) {
  return (*(int*)a - *(int*)b);
}

int main() {
  int array[] = {5, 2, 9, 1, 5, 6};
  int size = sizeof(array) / sizeof(array[0]);

  qsort(array, size, sizeof(int), compare);

  for (int i = 0; i < size; i++) {
    printf("%d ", array[i]);
  }
  printf("**\n**");

  return 0;
}
```

注意事项

- 使用 `malloc`、`calloc` 和 `realloc` 分配的内存必须使用 `free` 释放，以避免内存泄漏。
- `atoi`、`atol` 和 `atof` 函数不提供错误检查，使用时需确保输入格式正确。建议使用 `strtol`、`strtoul` 和 `strtod` 以进行更安全的转换。
- `qsort` 和 `bsearch` 函数需要自定义比较函数，以处理不同的数据类型和排序需求。

通过理解和使用 `<stdlib.h>` 提供的函数，可以在 C 程序中进行内存管理、进程控制、随机数生成、环境管理、数学转换以及排序和搜索等操作，从而编写更加灵活和功能丰富的程序。