

SysY 运行时库

SysY 运行时库提供一系列 I/O 函数、计时函数等用于在 SysY 程序中表达输入/输出、计时等功能需求。这些库函数不用在 SysY 程序中声明即可在 SysY 的函数中使用。需要指出的是：部分 SysY 库函数的参数类型会超出 Sys Y 支持的数据类型，如可以为字符串。SysY 编译器需要能处理这种情况，将 SysY 程序中这样的参数正确地传递给 SysY 运行时库。

1. SysY 运行时库的相关文件

大赛组委会提供如下 SysY 运行时库文件给参赛选手：

- **libsysy.a** 和 **libsysy.so** 分别是 SysY 运行时库的静态库和动态库文件(面向大赛的目标平台)。后续为公平起见，大赛组委会将统一按静态库链接进行评测。
- **sylib.h** 其中包含 SysY 运行时库涉及的函数等的声明。

注：在 SysY 源程序中不出现对 **sylib.h** 的文件包含，而由 SysY 编译器来分析和处理 SysY 程序中对这些函数的调用。

2. I/O 函数

SysY 运行时库提供一系列 I/O 函数，支持对整数、字符以及一串整数的输入和输出。为便于在 SysY 程序中控制输出的格式，诸如 `putf` 这样的 I/O 函数会使用超出 Sys Y 语言支持的数据类型的参数，如格式字符串。

SysY 运行时库提供如下的 I/O 函数，其中各个参数为整数值、变量、数组元素访问表达式：

1) `int getint()`

输入一个整数，返回对应的整数值。

示例：`int n; n = getint();`

2) `int getch()`

输入一个字符，返回字符对应的 ASCII 码值。

示例：`int n; n = getch();`

3) `int getarray(int [])`

输入一串整数，第 1 个整数代表后续要输入的整数个数，该个数通过返

回值返回；后续的整数通过传入的数组参数返回。

注：getarray 函数获取传入的数组的起始地址，不检查调用者提供的数组是否有足够的空间容纳输入的一串整数。

示例： `int a[10][10]; int n; n = getarray(a[0]);`

4) void putint(int)

输出一个整数的值。

示例： `int n=10; putint(n); putint(10); putint(n);`

将输出: 101010

5) void putch(int)

将整数参数的值作为 ASCII 码，输出该 ASCII 码对应的字符。

注：传入的整数参数取值范围为 0~255，putch()不检查参数的合法性。

示例： `int n=10; putch(n);`

将输出换行符

6) void putarray(int, int[])

第 1 个参数表示要输出的整数个数(假设为 N)，后面应该跟上要输出的 N 个整数的数组。putarray 在输出时会在整数之间安插空格。

注：putarray 函数不检查参数的合法性。

示例： `int n=2; int a[]={2,3}; putarray(n, a);`

输出： 2: 2 3

7) void putf (<格式串>, int, ...)

第 1 个参数为格式字符串，其中仅包含 2 种格式符，即‘%d’和‘%c’；该函数将根据格式串进行输出，遇到普通字符则原样输出，遇到格式符‘%d’或‘%c’则从第 2 个参数起依次取对应参数的值按整数或字符输出。

示例： `int n=2; int a[]={2,3};`

`putf(“%d: %d(%c), %d(%c)”, n, a[0], a[0]+48, a[1], a[1]+48);`

输出： 2: 2(2), 3(3)

3. 计时函数

SysY 运行时库提供 starttime、stoptime 一对函数用于对 SysY 中的一段代码的运行进行计时。在一个 SysY 程序中，可以插入多对 starttime、stoptime 调用来获得每对调用之间的代码的执行时长，并在 SysY 程序执行结束后得到这些计时的累计执行时长。需要注意的是：starttime、stoptime 不支持嵌套调用的形式，即不支持 starttime()...starttime()...stoptime()...stoptime()这样的调用执行序列。下面分别简介所提供的计时函数的访问接口：

8) void starttime()

开启计时器。此函数应和 stoptime()联用。

9) void stoptime()

停止计时器。此函数应和 starttime()联用。程序会在最后结束的时候，整体输出每个计时器所花费的时间，并统计所有计时器的累计值。

格式为 Timer#编号@开启行号-停止行号: 时-分-秒-微秒

示例:

```
void foo(int n){
    starttime();
    for(int i=0;i<n;i++)system("sleep 1");
    stoptime();
}

int main(){
    starttime();
    for(int i=0;i<3;i++)system("sleep 1");
    stoptime();
    foo(2);
}
```

输出:

Timer#001@0010-0012: 0H-0M-3S-3860us

Timer#002@0005-0007: 0H-0M-2S-2660us

TOTAL: 0H-0M-5S-6520us