

必应

你好，这是Bing。我可以告诉你如何传递参数给Python程序。 😊

如果你使用popen或者system函数，你可以直接在命令行中添加参数，就像在shell中一样。例如，如果你的Python程序是：

```
import sys
print("Hello from Python!")
print("The arguments are:", sys.argv)
```

你可以在C语言中这样调用它：

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    // 使用popen函数
    FILE *fp = popen("python test.py arg1 arg2", "r");
    if (fp == NULL) {
        printf("popen failed!\n");
        return -1;
    }
    char buffer[1024];
    while (fgets(buffer, sizeof(buffer), fp) != NULL) {
        printf("%s", buffer);
    }
    pclose(fp);
    // 使用system函数
    system("python test.py arg1 arg2");
    return 0;
}
```

这样，Python程序就可以接收到arg1和arg2作为参数。

如果你使用fork和execve函数，你可以在argv数组中添加参数，就像在main函数中一样。例如，你可以在C语言中这样调用Python程序：

```
#include <stdio.h>
#include <unistd.h>
int main() {
    // 使用fork和execve函数
    pid_t pid = fork();
    if (pid == -1) {
        printf("fork failed!\n");
        return -1;
    }
    if (pid == 0) {
        // 子进程
        char *argv[] = {"python", "test.py", "arg1", "arg2", NULL};
        char *envp[] = {NULL};
        execve("/usr/bin/python", argv, envp);
        printf("execve failed!\n");
        return -1;
    }
    else {
        // 父进程
        wait(NULL); // 等待子进程结束
        printf("Child process finished!\n");
        return 0;
    }
}
```

这样，Python程序也可以接收到arg1和arg2作为参数。