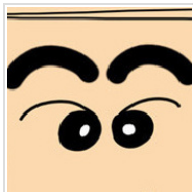


Nothing replaces hard work

目录视图 摘要视图 RSS 订阅

个人资料



lmx077

访问： 644819次
积分： 7300
等级： 5
排名： 第2516名

原创： 135篇
转载： 35篇
译文： 7篇
评论： 114条

文章搜索

声明

本博客乃学习笔记，没有纯粹无意义的转载。作者除了对自己负责，不对任何读者负责。欢迎指出文章错误，如果原意交朋友，可以通过Gmail联系我 (mingxinglai@gmail.com)，博客基本不再更新，欢迎访问我的独立博客 <http://mingxinglai.com>

文章分类

C++/C (25)
算法 (36)
Unix系统编程 (9)
OJ (26)
VC/MFC (8)
笔试面试 (31)
linux/shell (19)
STL (6)
IT杂文 (9)
编程珠玑 (9)
Python (1)
vim (6)

文章存档

[【公告】关于开启用户注册及登录手机短信验证的通知](#) [CSDN日报20170412 —— 《程序员该如何定位？看这四大方向》](#) [博客搬家，有礼相送](#)

Unix编程之size_t、ssize_t

标签： [unix](#) [编程](#) [extension](#) [freebsd](#) [编译器](#) [types](#)

2012-04-04 20:17 27365人阅读 评论(0)

分类：

[Unix系统编程 \(8\)](#)

版权声明：本文为博主原创文章，未经博主允许不得转载。

对于赶时间的朋友，只看第一段就好了。

首先，我非常肯定以及确定的告诉你ssize_t是有符号整型，在32位机器上等同与int，在64位机器上等同与long int，有没有注意到，它和long数据类型有啥区别？其实是一样的。size_t就是无符号型的ssize_t，也就是unsigned long/ unsigned int (在32位下)，不同的编译器或系统可能会有区别，主要是在32位机器上int和long是一样的。在64位没有测试，但是参见百度百科，size_t在64位下是64位，那么size_t的正确定义应该是typedef unsigned long size_t。

ssize_t:

前一阵子，上Unix程序设计，问老师说ssize_t到底是什么数据类型，老师说，你直接用就好了，真要知道是什么数据类型，可以去types.h里去看一下，回去以后找了一下没有找到就放弃了，今天写代码又遇到这个问题，我觉得不是办法，还是得弄清楚它到底是什么类型。当然了，对于ssize_t到底是什么数据类型，不知道问题也不大，但是对于控制欲强和好奇心重的人，总是想知道的，而且我相信，知道了总会有好处的，写代码的时候心里有底。

查找步骤：

1. 包含头文件的时候是这么做的：`#include <sys/types.h>`，执行locate types.h跳出一堆文件，这也是为什么我第一次尝试失败的原因。首先执行locate sys/types.h,就可以找到types.h的存放位置了。

2. 打开types.h，搜索 ssize_t,得到如下结果

```
#ifndef __ssize_t_defined
typedef __ssize_t ssize_t;
#define __ssize_t_defined
```

3. 再搜索__ssize_t就找不到任何有价值的信息了，观察他头文件，也就是#include <bits/types.h>，再次打开bits/types.h

4. 搜索__ssize_t找到了这么一条语句：__STD_TYPE __SSIZE_T_TYPE __ssize_t;这

索__STD_TYPE_SSIZE_T_TYPE 又找到了这样一句话：
/* We want __extension__ before typedef's that use nonstan
such as 'long long' in C89 mode. */

```
# define __STD_TYPE __extension__ typedef
```

也就是说，__STD_TYPE __SSIZE_T_TYPE __ssize_t;这
__SSIZE_T_TYPE __ssize_t，那么重点就是找到__SSIZE

关闭

立即抢购

2012年10月 (5)
2012年09月 (2)
2012年08月 (1)
2012年07月 (1)
2012年06月 (14)

展开

阅读排行

Shell脚本之循环 (41768)
将两个有序链表合并成一 (35368)
Unix编程之size_t、ssize (27362)
MFC连接mysql数据库 ((23630)
深度优先遍历算法的非递 (20412)
[编程之美]二叉树的层次 (19577)
10.1动态规划例题：数字 (14142)
遗传算法求解旅行商问题 (13173)
MFC对话框画图 (11973)
个性化vim之映射(map,vr (11939)

评论排行

[编程珠玑读书笔记]快速 (9)
MFC连接mysql数据库 ((6)
Unix编程之size_t、ssize (5)
[编程之美]二叉树的层次 (5)
深度优先遍历算法的非递 (5)
MFC对话框画图 (4)
5.3日历问题 (4)
一道有趣的面试题——将 (4)
迪杰斯特拉算法 (4)
[算法导论读书笔记]Dijkstra (4)

推荐文章

* 【《Real-Time Rendering 3rd》提炼总结】(一) 全书知识点总览
* CSDN日报20170409 —— 《扯蛋的密码规则》
* Shader2D: 一些2D效果的Shader实现
* 一个屌丝程序员的人生 (六十一)
* 自定义控件三部曲视图篇 (三) ——瀑布流容器 WaterFallLayout实现
* 面向服务的体系架构 (SOA) —架构篇

最新评论

MFC连接mysql数据库 (十分钟排 A1200266: 请问有没有客户端-服务端的例子, 目前客户端代码中包含了mysql的账号信息, 内存搜索或反编译都能看到...
MFC连接mysql数据库 (十分钟排 A1200266: 请问有没有客户端-服务端的例子, 目前客户端代码中包含了mysql的账号信息, 内存搜索或反编译都能看到...
python语言入门
qq_34083182: 蟹蟹
将两个有序链表合并成一个有序

__SIZE_T_TYPE, 又没有什么有价值的信息了, 与步骤3类似, 我们又找到了这条语句: #include <bits/typesizes.h> , 打开typesizes.h文件, 搜索 __SSIZE_T_TYPE, 很轻松的就找到了#define __SSIZE_T_TYPE __SWORD_TYPE, 也就是说最开始的 __STD_TYPE __SSIZE_T_TYPE __ssize_t; 这条语句其实就是 typedef __SWORD_TYPE __ssize_t, typesize.h文件里找不到 __SWORD_TYPE, 于是我们又回到了bit/types.h中, 结果终于出现了。

5.在bit/types.h中很轻松的就找到了下面这几行语句。

```
#if __WORDSIZE == 32
# define __SQUAD_TYPE      __quad_t
# define __UQUAD_TYPE      __u_quad_t
# define __SWORD_TYPE      int
# define __UWORD_TYPE      unsigned int
# define __SLONG32_TYPE    long int
# define __ULONG32_TYPE    unsigned long int
# define __S64_TYPE        __quad_t
# define __U64_TYPE        __u_quad_t
/* We want __extension__ before typedef's that use nonstandard base types
   such as 'long long' in C89 mode. */
# define __STD_TYPE        __extension__ typedef
#elif __WORDSIZE == 64
# define __SQUAD_TYPE      long int
# define __UQUAD_TYPE      unsigned long int
# define __SWORD_TYPE      long int
# define __UWORD_TYPE      unsigned long int
# define __SLONG32_TYPE    int
# define __ULONG32_TYPE    unsigned int
# define __S64_TYPE        long int
# define __U64_TYPE        unsigned long int
/* No need to mark the typedef with __extension__ */
# define __STD_TYPE        typedef
#else
# error
#endif
```

我们几乎都可以猜到 __WORDSIZE 是什么意思了, 它的定义在 #include <bits/wordsize.h> 中, 表示计算机系统是几位的。从上面我们知道, 在32位计算机系统中, ssize_t 是 int 型, 占4个字节, 在64位计算机系统中, ssize_t 是 long int 型, 占8个字节。

size_t:

晕, size_t 的资料居然找了半天。size_t 大家都知道, 是无符号整型, 在 /usr/include/i386-linux-gnu/sys/stddef.h 下看到这么几条语句:

```
/* __size_t is a typedef on FreeBSD 5!, must not trash it. */
#else
#define __size_t
#endif
#ifndef __SIZE_TYPE__
#define __SIZE_TYPE__ long unsigned int
#endif
#if !(defined (__GNU__) && defined (size_t))
typedef __SIZE_TYPE__ size_t;
#endif
#ifdef __BEOS__
typedef long ssize_t;
```

可以肯定的是, size_t 是无符号整型, 至于是 long 型, 还是 int 型, 可能不同的编译器有不同的定义, 我这里没有64位的机器, 无法验证。这也验证了我们上面所说的 ssize_t 其实

关闭

山羊哥哥: 赞
 POJ并查集的题目汇总
 Deny_小田: 怒赞 ^^
 10.1动态规划例题: 数字三角形
 chenpkai: 楼主在描述递归解法重复计算时有误: "那么, 每次计算MaxSum(r, j)的时候, 都要计算一次 M...
 将两个有序链表合并成一个有序链
 xfyae: 思路简明扼要, 很清晰, 尤其递归, 写的太好了!
 python语言入门
 fa00x001: 写的非常好, 写的很细, 很全。
 [编程之美]二叉树的层次遍历
 zzhznx: if root is None: return []
 q = Q...
 Unix编程之size_t、ssize_t
 wuxianglonghaohao: 楼主是有心人, 这样深入挖掘的态度值得学习, 赞一个

就是一个long。

下面的是一点附加资料, 证明我找到的是正确的。:-)

一个基本的无符号整数的C / C ++类型。它是sizeof操作符返回的结果类型。该类型的大小是选择, 因此, 它可以存储在理论上是可能的任何类型的数组的最大大小。在32位系统为size_t将采取32位和64位 - 64位。

参考资料: <http://baike.baidu.com/view/3236587.htm>

Data Type: ssize_t

This data type is used to represent the sizes of blocks that can be read or written in a single operation. It is similar to size_t, but must be a signed type.

参考资料: http://www.delorie.com/gnu/docs/glibc/libc_239.html

顶 9
 踩 1

上一篇 Unix编程之环境变量

下一篇 将两个有序链表合并成一个有序链表

我的同类文章

Unix系统编程 (8)

- [unix系统编程]读者-写者问题 2012-05-27 阅读 1493
- perror和strerror的区别 2012-05-10 阅读 9946
- [Unix系统编程]用信号量实现哲学家... 2012-05-10 阅读 1758
- 文件互斥的方式实现哲学家... 2012-05-05 阅读 833
- 测试管道的容量 2012-04-28 阅读 1123
- Testing the Type of a File in... 2012-04-24 阅读 727
- Unix编程之环境变量 2012-04-04 阅读 1196
- Unix下文件的访问权限 2012-03-17 阅读 963

猜你在找

- Ceph分布式存储系统的另一个选择
- linux操作系统下c语言编程入门
- 顾荣: 开源大数据存储系统Alluxio (原Tachyon) 的
- 经典linux下c编程
- C++ 单元测试 (GoogleTest)
- linux操作系统下c语言编程入门
- python项目之学员CRM管理系统开发阶段二
- APUE Linux C 编程开发环境工具链编译链接接库基
- 4.7. 存储类&作用域&生命周期&链接属性-C语言高级专
- UNIX网络编程卷2-读书笔记

查看评论

关闭

5楼 wuxianglonghaohao 2016-06-27 10:11发表




楼主是有心人, 这样深入挖掘的态度值得学习, 赞一个

4楼 [游客] 2016-05-12 11:18发表




kscope之类的工具能简化lz的工作, 大概


3楼 haohaibo031113 2016-01-20 16:07发表

 分析的很细致~\(\ge\ \nabla\ \le\)/~

2楼 [i龙家小少](#) 2013-02-25 21:27发表

 佩服下楼主的判断.....好清晰的逻辑！不过我想说.....我在vim中看代码时看到ssize_t，因为有定义过tag，所以Ctrl+T，一路就能找到对应的ssize_t定义，typedef __kernel_ssize_t ssize_t; 同样方法找__kernel_ssize_t可以找到typedef long __kernel_ssize_t;呵呵，方法简单点，适合初学者

1楼 [tjs2200120](#) 2013-02-20 10:44发表


 感谢楼主耐心讲解，学习了

发表评论

用户名：

uudou

评论内容：



提交

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题

Hadoop

AWS

移动游戏

Java

Android

iOS

Swift

智能硬件

Docker

OpenStack

VPN

Spark

ERP

IE10

Eclipse

CRM

JavaScript

数据库

Ubuntu

NFC

WAP

jQuery

BI

HTML5

Spring

Apache

.NET

API

HTML

SDK

IIS

Fedora

XML

LBS

Unity

Splashtop

UML

components

Windows Mobile

Rails

QEMU

KDE

Cassandra

CloudStack

FTC

coremail

OPhone

CouchBase

云计算

iOS6

Rackspace

Web App

SpringSide

Maemo

Compuware

大数据

aptech

Perl

Tornado

Ruby

Hibernate

ThinkPHP

HBase

Pure

Solr

Angular

Cloud Foundry

Redis

Scala


Django

Bootstrap

公司简介 | 招贤纳士 | 广告服务 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 |

江苏乐知网络技术有限公司

京 ICP 证 09002463 号 | Copyright © 1999-2016, CSDN.NET, All Rights Reserved 

关闭