

嵌入式系统测试

主要测试业务范围

- * 各军种、兵种武器系统、指挥系统、决策系统
- * 通讯系统、导航系统、网络系统
- * 医疗系统
- * 工控系统等

嵌入式软件的特点

- * 实时性：实时约束、实时控制
- * 依赖于特定硬件环境：CPU和硬件体系多样而难以使用固定测试设备
- * 交叉式开发：基于Host/Target方法
- * 外部环境交互作用：测试工具必须同开发环境相配套
- * 一般资源缺乏：结构紧凑而难以插装

测试指标

静态分析

- 软件质量评价
- 结构分析
- 代码走查
- 功能确认

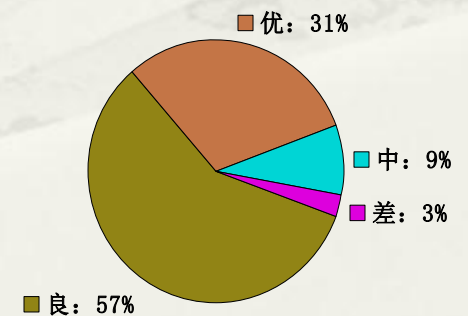
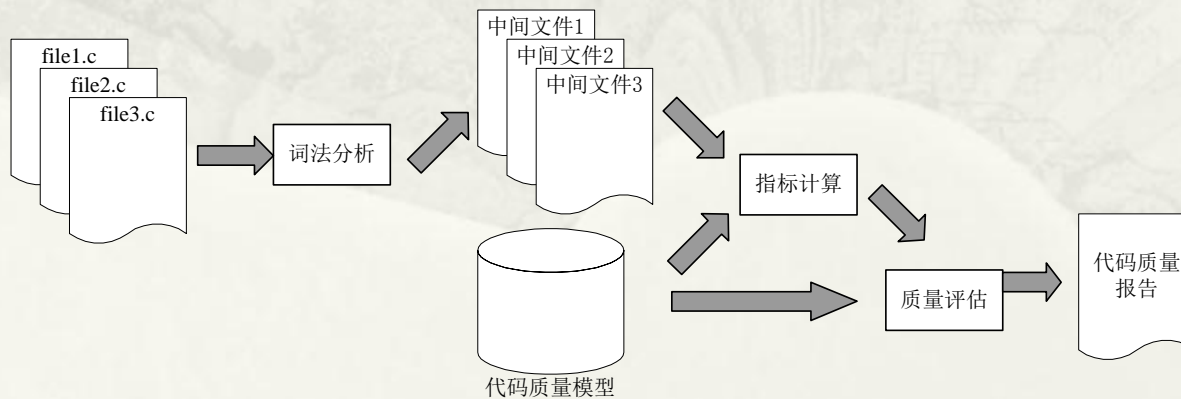
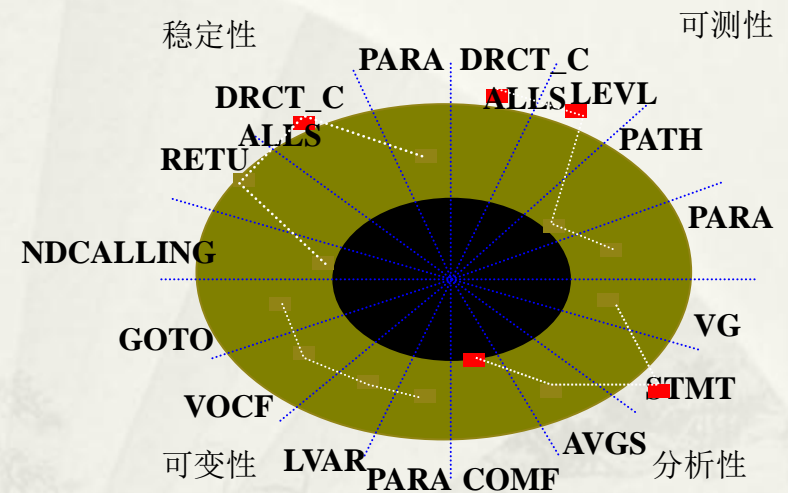
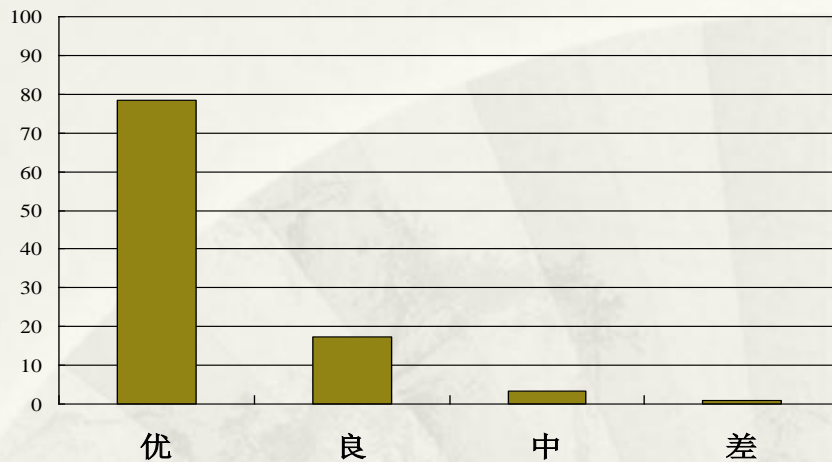
动态分析

- 覆盖率分析
- 内存分析
- 执行性能
- 代码跟踪

测试指标 - 软件质量评价

- * 软件质量可以由质量复杂度来度量，这些复杂度包括：圈复杂度 $V(g)$ 、强化复杂度、设计复杂度 $iV(g)$ 、集成复杂度、代码行数、Halstead复杂度；
- * 软件质量评价通过计算复杂度，监视不同时期软件的质量，提供详细的报告，帮助质量部门提高软件的质量；
- * 遵循ISO 9126、DO-178B、SEI/CMM标准；
- * 所有的报告都是可定制的、图形化的。

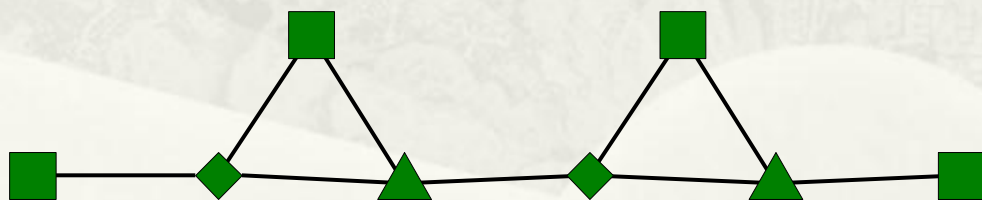
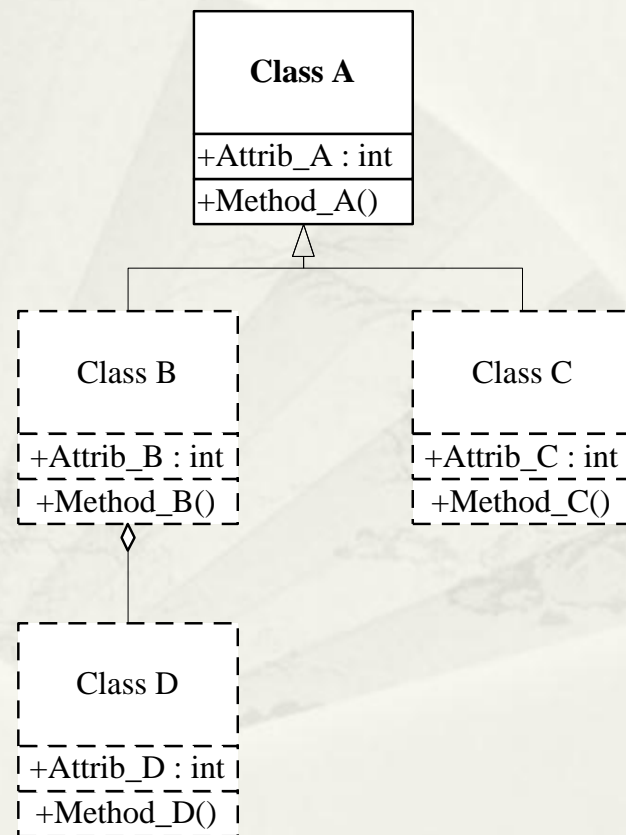
软件质量评价模型



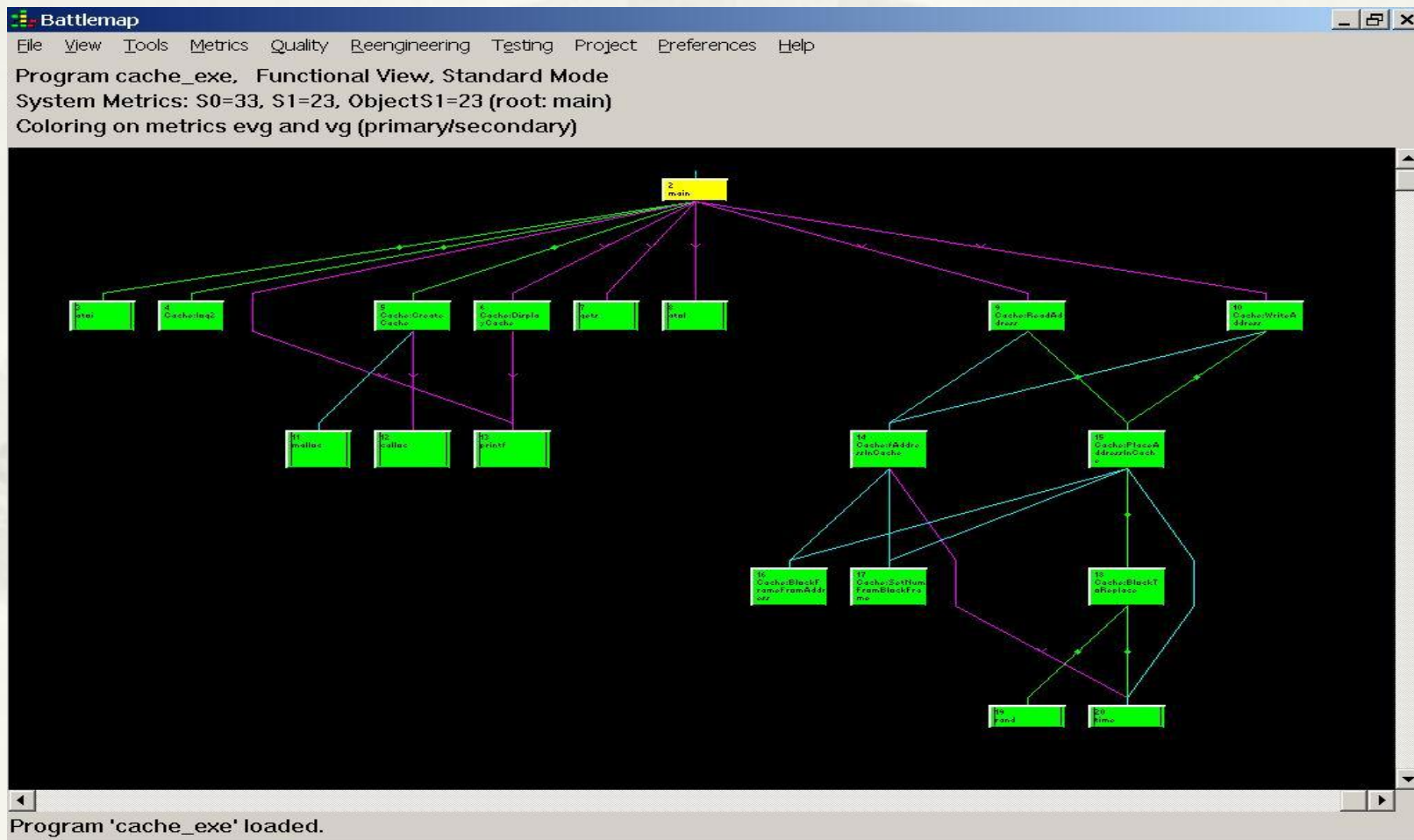
测试指标 - 结构分析

- * 分析程序的源代码，建立起强大的数据库，然后生成一个以颜色来标识的系统结构图、函数调用关系图；
- * 这些结构图为质量控制经理和开发者之间提供了一个准确的沟通点；
- * 而且有了这个图形就很容易指出系统中的高危险区域，并可为以后进行的重点测试提供依据。

结构分析1



结构分析2



结构分析3

Battlemap

File View Tools Metrics Quality Reengineering Testing Project Preferences Help

Program cache_exe, Functional View, Standard Mode
System Metrics: S0=33, S1=23, ObjectS1=23 (root: main)
Coloring on metrics evg and vg [primary/secondary]

Graph/Listing for 'Cache:DisplayCache'

Zoom In Zoom Out Print... Save As... Save Text... Close Help... Graph (45%):

Magnification Level: 2
Page 1 of 3 Cache:DisplayCache

Program: cache_exe
Cache:DisplayCache (C)
Cyclomatic Graph
Cyclomatic 3
Essential 1
Design 3

05/20/02
Superimposed
Upward Edges
Loop Edges
Plain Edges

0
1*
2
3
4
5
6*
7
8
9
10
11*
12
13
14
15

	C	Cache:DisplayCache	v(G)	ev(G)	iv(G)
			3	1	3

Program : cache_exe
File : E:\Cache\Cache.c
Language: cw_C_inst
Module Module
Letter Name v(G) ev(G) iv(G)

```

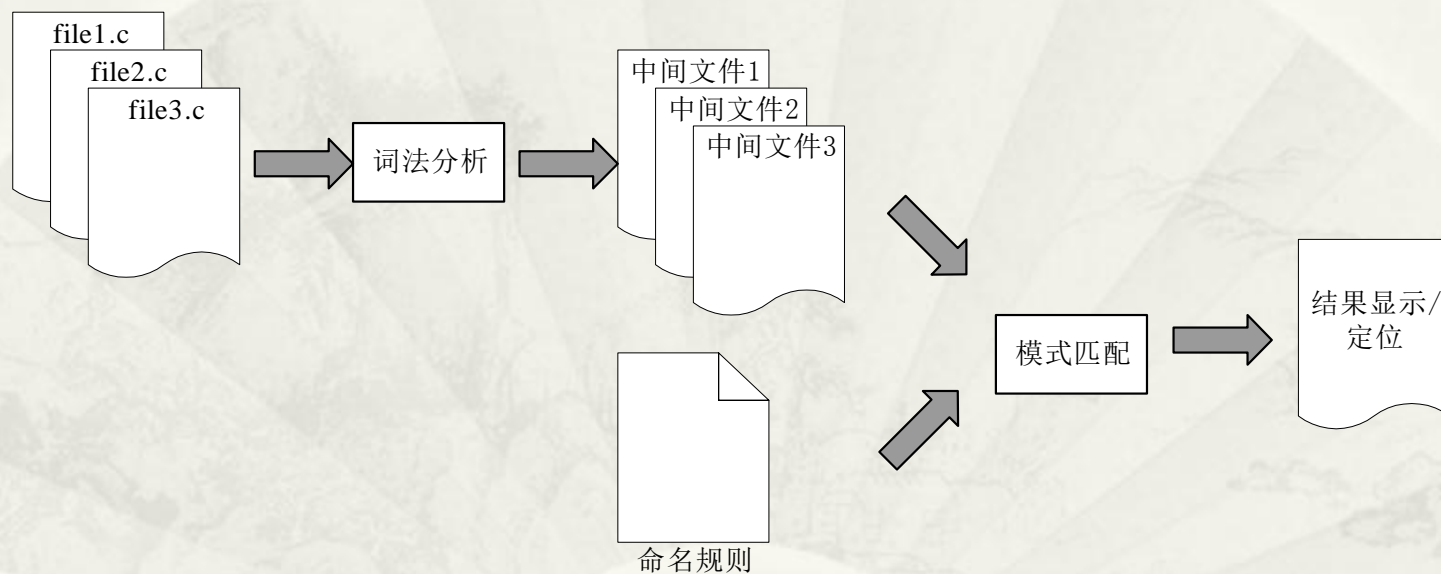
88      C0      DisplayCache(cache)
89      Cache *cache;
90      {
91      /* print out the cache */
92
93      unsigned int i, j;
94
95      C1* C2      (void)printf("Sets in cache: %u\n", num
96      C3 C4 C5      for (i = 0; i < num_sets_in_cache; i++)
97      C6* C7      (void)printf(" Set %u: %u blocks\n",
98      C8 C9 C10      for (j = 0; j < num_blocks_per_set
99      C11* C12      (void)printf(" Block %u: Val:
100     cache->sets[i].blocks[j]
101     cache->sets[i].blocks[j]
102     cache->sets[i].blocks[j]

```

测试指标 - 代码走查

- * 根据为项目定制的规则自动的检查代码编程，可以避免错误陷阱和代码错误；
- * 用所选的规则对源代码进行验证，指出所有不符合编程规则的代码，并提出改进源代码的解释和建议；
- * 通过文本编辑器直接访问源代码并指出需要纠正的位置；
- * 可生成HTML格式的代码规则的审核报告，供软件团队成员参考。

代码走查 - 命名规则检查



MyIE - [C++Rule/C++Rule_main.html]

文件(E) 编辑(E) 查看(V) 收藏(A) 快捷(C) 选项(O) 工具(T) 窗口(W) 帮助(H)

地址 file:///G:/C++Rule/C++Rule_main.html#ViolationsRule

Hangman.cpp

Rule Mnemonic	Rule Name	State	Lines
constrdef	Default constructor	Not Violated	-
classuse	Hidden class uses	Not Violated	-
identfmt	Identifier format	Not Violated	-
operass	Assignment operator	Not Violated	-
identres	Reserved identifiers	Not Violated	-
slcom	Use // comments	Not Violated	-
ctrlblock	Blocks in control statements	Violated	52, 70, 80, 81, 83, 115, 137, 148, 229,
headercom	Function and class header comments	Violated	11, 21, 25, 36, 58, 63, 68, 74, 88, 93, 100, 105, 120, 125, 129, 145, 151, 156, 161, 166, 225,
mfunc	Inline functions instead of macro functions	Not Violated	-
swdef	default within switch	Not Violated	-

Panorama pr http://www. Download th Download th C++Rule

99M 5

测试指标 - 功能确认

- * 根据软件概要设计说明测试软件功能模块的功能；
- * 检验软件单元之间的接口关系，并把经过测试的单元构成符合设计要求的软件。

测试指标 - 覆盖率分析

- * 提高软件的可靠性；
- * 语句覆盖率SC (Statement Coverage) : 行覆盖、段覆盖、基本块覆盖
- * 决策覆盖率DC (Decision Coverage)
- * 多条件决策覆盖率MCDC (Modified Condition / Decision Coverage)

覆盖率分析 - 基本块覆盖度

Coverage Data [0] - (HWIC [192.168.2.33])

File View Help

Row	Function Name	File Name	% Coverage
6	getLfork	Philosopher.c	100.00%
7	getLFork	Philosopher.c	100.00%
8	iround	Philosopher.c	100.00%
9	clear_fork	dinPhilMain.c	100.00%
10	draw_fork	dinPhilMain.c	100.00%
11	draw_eating	dinPhilMain.c	100.00%
12	draw_waiting	dinPhilMain.c	100.00%
13	draw_thinking	dinPhilMain.c	100.00%
14	position_flush	dinPhilMain.c	100.00%
15	position	dinPhilMain.c	100.00%
16	cls	dinPhilMain.c	100.00%
17	stepFSM	Philosopher.c	95.00%
18	randMsg	Philosopher.c	91.67%
19	philosopherTask	Philosopher.c	90.91%
20	vxmain	dinPhilMain.c	85.19%
21	queueAdd	Queue.c	75.00%
22	hasBothForks	Philosopher.c	75.00%
23	createForks	dinPhilMain.c	62.50%
24	createPhilosopher	Philosopher.c	60.00%
25	Memerr	Philosopher.c	57.14%
26	hasNoForks	Philosopher.c	50.00%
27	hasRForkOnly	Philosopher.c	50.00%
28	hasLForkOnly	Philosopher.c	50.00%
29	pollQueue	Philosopher.c	42.86%

37 of 37 rows Function Coverage SC Block Coverage - 69.20%

覆盖率分析 - 测试结果

```
Source File: /svt1/svt/ctdemo/peterk/lib/ctdemo.c

void
cmdReceivedISR( void )
{
    CmdHdr cmd;
    AMCUserTag( BEG_CMD_PROCESSING );
    readAnyCmd( &cmd );
    if (!( cmd.idata ))
    {
        switch ( cmd.cmd )
        {
            case startMeasurementE:
            {
                startMeasurement();
                break;
            }
            case stopMeasurementE:
            {
                stopMeasurement();
                break;
            }
            case continueMeasurementE:
            {
                continueMeasurement();
                break;
            }
            case queryStatusE:
            {
                queryStatus();
                break;
            }
            default:
            {
                AMCPrintf("ERR:cmdReceivedISR(): unknown no idata cmd: %d\n", cmd.cmd );
            }
        }
    }
    else
    {
        switch ( cmd.cmd )
        {
            case startMeasurementE:
            {
                startMeasurement();
                break;
            }
            case stopMeasurementE:
            {
                stopMeasurement();
                break;
            }
            case continueMeasurementE:
            {
                continueMeasurement();
                break;
            }
            case queryStatusE:
            {
                queryStatus();
                break;
            }
            default:
            {
                AMCPrintf("ERR:cmdReceivedISR(): unknown no idata cmd: %d\n", cmd.cmd );
            }
        }
    }
}

** These commands require inputs.
...

```

高亮显示执行过的代码

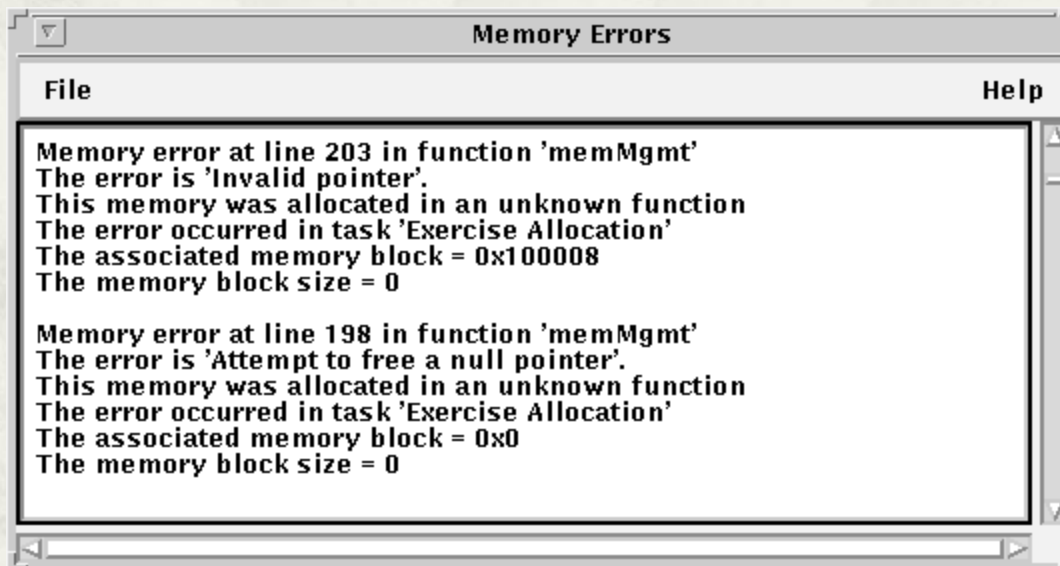
重点检查部分函数，审查未执行过的代码行的执行条件

Close Help

测试指标 - 内存分析

- * 指针和泄露错误：包括接口泄露、内存泄露、未分配的指针错误
- * 内存错误：包括内存分配冲突、栈空间溢出、动态存储溢出、静态存储溢出、无效的句柄被锁定、句柄没有被锁定
- * API和OLE错误：包括API函数返回失败、API函数未执行、无效的变量（包括指针变量、字符串变量等）、OLE接口方法的变量无效、OLE接口方法失败、线程调用库函数错误

内存分析 - 内存错误

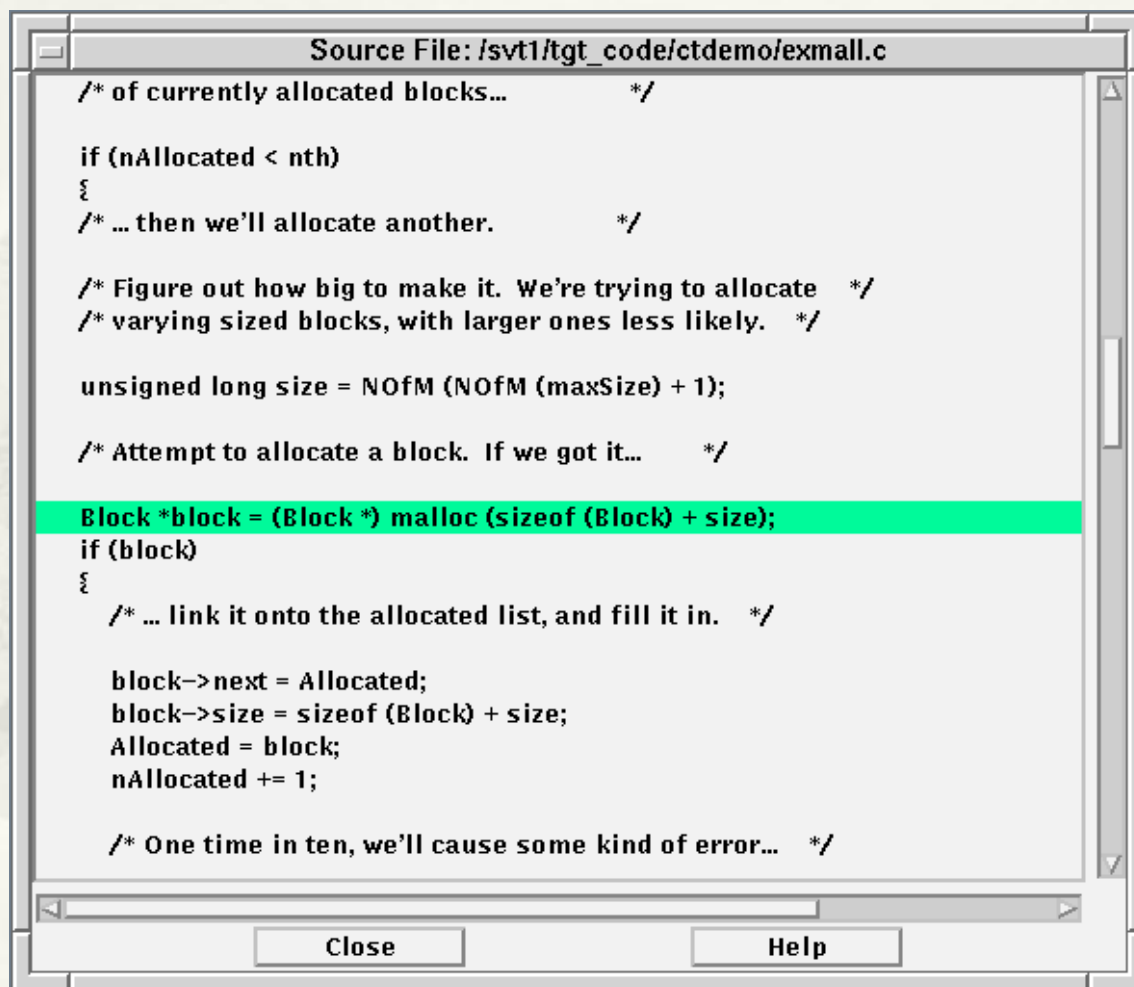


```
Memory Errors
File Help

Memory error at line 203 in function 'memMgmt'
The error is 'Invalid pointer'.
This memory was allocated in an unknown function
The error occurred in task 'Exercise Allocation'
The associated memory block = 0x100008
The memory block size = 0

Memory error at line 198 in function 'memMgmt'
The error is 'Attempt to free a null pointer'.
This memory was allocated in an unknown function
The error occurred in task 'Exercise Allocation'
The associated memory block = 0x0
The memory block size = 0
```

内存分析 - 相应的源代码



```
Source File: /svt1/tgt_code/ctdemo/exmall.c

/* of currently allocated blocks... */

if (nAllocated < nth)
{
/* ... then we'll allocate another. */

/* Figure out how big to make it. We're trying to allocate */
/* varying sized blocks, with larger ones less likely. */

unsigned long size = NOFM (NOFM (maxSize) + 1);

/* Attempt to allocate a block. If we got it... */

Block *block = (Block *) malloc (sizeof (Block) + size);
if (block)
{
/* ... link it onto the allocated list, and fill it in. */

block->next = Allocated;
block->size = sizeof (Block) + size;
Allocated = block;
nAllocated += 1;

/* One time in ten, we'll cause some kind of error... */
```


测试指标 - 执行性能

- * 为软件的性能改善和优化提供依据；
- * 测试所有函数执行时间和调用次数；
- * 测试所有任务执行时间和调用的函数。

执行性能 - 任务性能

Task Data [0] - (HWIC [192.168.2.33])

File View Help




Row	Name	# Instances	# Entries	Minimum(u...	Maximum(...	Average(us)	Cumulative(us)	% Total Time
0	s1u0a	1	1	1.3835058...	0.0	65,763,74...	65,763,742.1	17.64%
1	tExcTaska	1	1	327.7	327.7	327.7	327.7	0.00%
2	tNetTaska	1	2,203	212.3	699.0	261.0	574,908.2	0.15%
3	tOnea	1	3,830	69.4	16,631.9	15,994.8	61,260,262.7	16.43%
4	tTwoa	1	3,865	72.5	19,305.4	15,858.7	61,293,819.2	16.44%
5	tivea	1	3,856	62.5	18,900.7	15,871.6	61,200,964.0	16.41%
6	tmera	1	1,142	13.0	1,772.6	542.2	619,180.6	0.17%
7	toura	1	3,856	60.2	18,695.7	15,877.6	61,224,002.6	16.42%
8	treea	1	3,839	70.8	18,141.9	15,882.5	60,972,987.5	16.35%

执行性能 - 函数性能

Performance Data [0] - (HWIC [192.168.2.33])

File View Help



Row	Function Name	# of Calls	Minimum(us)	Maximum(us)	Average(us)	Cumulative(u...	% Total Time
0	cls	4,950,037	13.1	3,032.6	21.7	107,237,689.4	69.90%
1	randMsg	4,949,729	4.4	2,290.1	8.0	39,602,298.0	25.81%
2	queueCount	4,905,389	0.6	325.4	0.6	3,104,274.0	2.02%
3	position	4,244	282.9	621.9	339.2	1,439,732.8	0.94%
4	position_flush	2,656	282.9	542.3	311.0	826,068.9	0.54%
5	stepFSM	3,142	47.7	329.6	101.0	317,469.9	0.21%
6	queueRemove	3,142	57.7	187.8	76.2	239,370.3	0.16%
7	queueAdd	3,142	39.7	64.1	45.6	143,213.9	0.09%
8	draw_fork	792	133.4	244.0	152.8	121,034.5	0.08%
9	clear_fork	788	133.7	238.6	149.5	117,830.9	0.08%
10	iround	84,323	0.6	119.9	0.6	54,514.3	0.04%
11	draw_waiting	641	71.7	165.6	81.6	52,328.3	0.03%
12	pollQueue	3,142	8.6	80.0	10.1	31,587.2	0.02%
13	releaseForks	639	21.7	130.8	44.0	28,090.0	0.02%
14	getLFork	641	11.3	100.5	20.4	13,054.4	0.01%
15	draw_thinking	218	73.2	148.1	81.3	17,723.1	0.01%
16	getRFork	641	11.6	99.8	20.4	13,103.3	0.01%
17	getBothForks	641	27.2	110.1	35.3	22,597.0	0.01%
18	draw_eating	215	72.4	147.4	80.6	17,324.3	0.01%
19	printState	1,069	6.5	67.4	7.6	8,074.1	0.01%
20	createPhilosopher	5	151.1	159.9	154.4	772.0	0.00%
21	queueInit	5	15.8	16.1	15.9	79.5	0.00%
22	Memerr	5	114.7	130.4	119.2	596.0	0.00%
23	FSMInit	5	4.5	8.0	5.2	26.6	0.00%

30 of 30 rows Hide Unknown Function

支持的处理器

- * ARM: ARM7、ARM9
- * IDT: MIPS R3000, 4640, 4700, 4600, 4650, 5000
- * Intel: StrongARM、80196 family、80Cx86/88 Family、80386 EX、80486DX, DX2, DX4, SX, SX2, SXL, SL、80960 family、Pentium II/III/4

- * AMD: Am186/188EM/ED/ER/ES、Am486 DX, DX2, DX4, DX5, SX, SX2、ELAN SC300、ELAN SC520
- * Motorola: 68000 Family、DragonBall (68EZ328, 68VZ328)、ColdFire、MPC509/555、PowerPC (PowerQUICC)、PowerPC (PowerQUICC II Family)、DSP56800 Family、M.core、8 or 16bit Microcontroller

支持的操作系统

- * Windows9X/NT/2000
- * Unix/Linux
- * RTOS: VxWorks、pSOS、Nucleus、DeltaOS、Hopen

支持的测试模式

- * 主机
- * 目标机
- * 模拟器
- * 仿真器

测试工具

- * 质量评估：Logiscope、McCabe
- * 单元测试：VectorCAST、Cantata、Attol
- * 集成测试：CodeTest
- * 系统测试：CodeTest
- * 其他：C++Test、NuMega DevPartner Studio

谢谢!