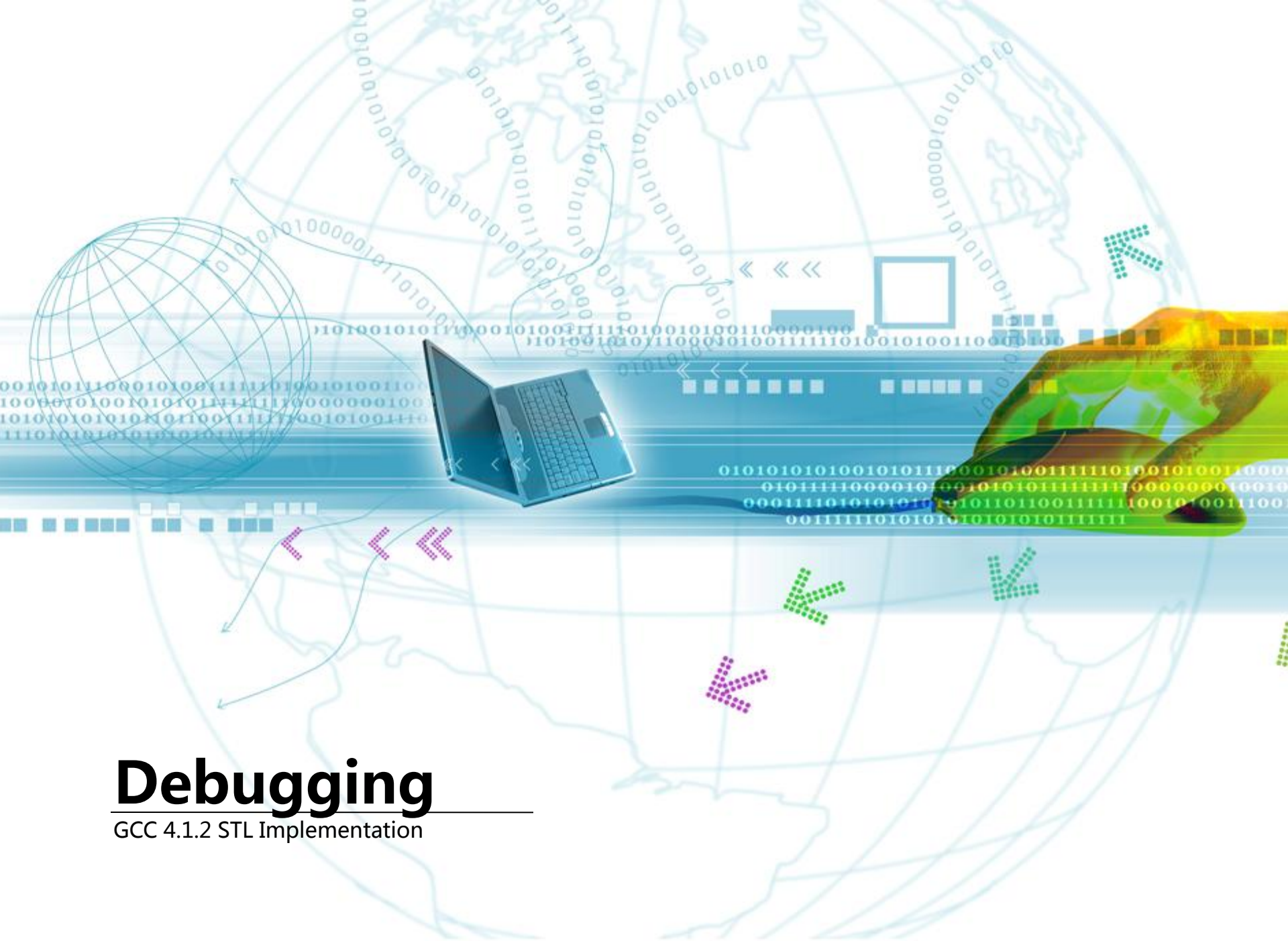


Debugging

GCC 4.1.2 STL Implementation

String

```c
#include <stdlib.h>

int main()
{
    char* p = "abcdefghigklmn";
    return 0;
}
```

1. 用help x观察x命令的使用格式
2. 用恰当的格式观察p

Pointer Array

```c
char* randomString();

int main()
{
    char* pointer_arr[N];
    for (int i = 0; i < N ; i++) {
        pointer_arr[i] = randomString();
    }

    for (int i = 0; i < N ; i++) {
        free(pointer_arr[i]);
    }
    return 0;
}
```

```c
char* randomString()
{
    int n = rand() % 10 + 1;
    char* buffer = (char*)malloc(n + 1);
    buffer[n] = '\0';

    for (int i = 0; i < n; ++i) {
        buffer[i] = 'a' + rand() % 26;
    }
    return buffer;
}
```

练习一

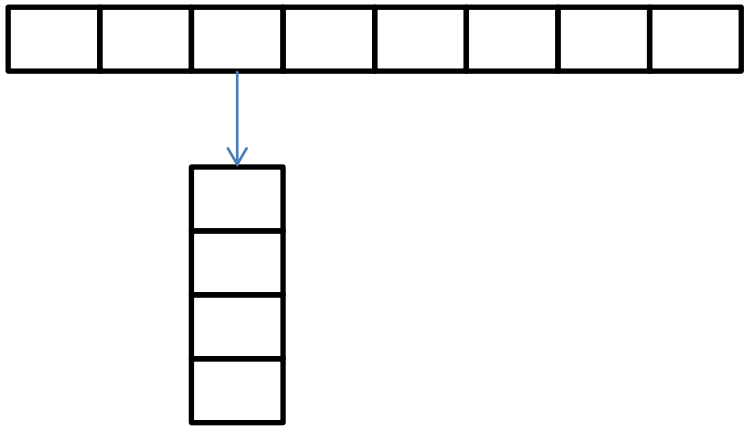
- 调试pointer_array，观察生成的随机字符串

Sample Array

```
typedef struct
{
    int     intMember;
    short   shortMember;
    int     len;
    char*   buffer;
}Sample;
```

```
int main()
{
    srand((unsigned)time(NULL));

    Sample* samples[N];
    for (int i = 0; i < N; ++i) {
        samples[i] = createSamples(randomString(),
            rand() % 1000, rand() % 1000);
    }

    for (int i = 0; i < N; i++) {
        destorySample(samples[i]);
    }
    return 0;
}
```

samples



练习二

- 调试sample_array写出生成的10个Sample变量

练习三

- 找出第99个学生的信息

```
typedef struct
{
    char* name;
    int age;
    char* address;
    int id;
}Student;
```

```
int main()
{
    srand((unsigned)time(NULL));

    Student* students[100] ;
    for (int i = 0; i < 100; ++i) {
        students[i] = createStudent();
    }

    for (int i = 0; i < 100; ++i) {
        destoryStudent(students[i]);
    }
    return 0;
}
```

分析xmlDoc内存结构

```
typedef struct _xmlDoc xmlDoc;
typedef xmlDoc *xmlDocPtr;
struct _xmlDoc {
    void                *_private;
    xmlElementType      type;
    char                *name;
    struct _xmlNode     *children;
    struct _xmlNode     *last;
    struct _xmlNode     *parent;
    struct _xmlNode     *next;
    struct _xmlNode     *prev;
    struct _xmlDoc      *doc;
    int                 compression;
    int                 standalone;
    struct _xmlDtd      *intSubset;
    struct _xmlDtd      *extSubset;
    struct _xmlNs       *oldNs;
    const xmlChar       *version;
    const xmlChar       *encoding;
    void                *ids;
    void                *refs;
    const xmlChar       *URL;
    int                 charset;
    struct _xmlDict     *dict;
    void                *psvi;
    int                 parseFlags;
    int                 properties;
};
```

```
int main()
{
    xmlDocPtr doc = xmlNewDoc(BAD_CAST"1.0");
    xmlNodePtr root_node =
        xmlNewNode(NULL, BAD_CAST"root");

    xmlDocSetRootElement(doc, root_node);

    xmlFreeDoc(doc);
    return 0;
}
```

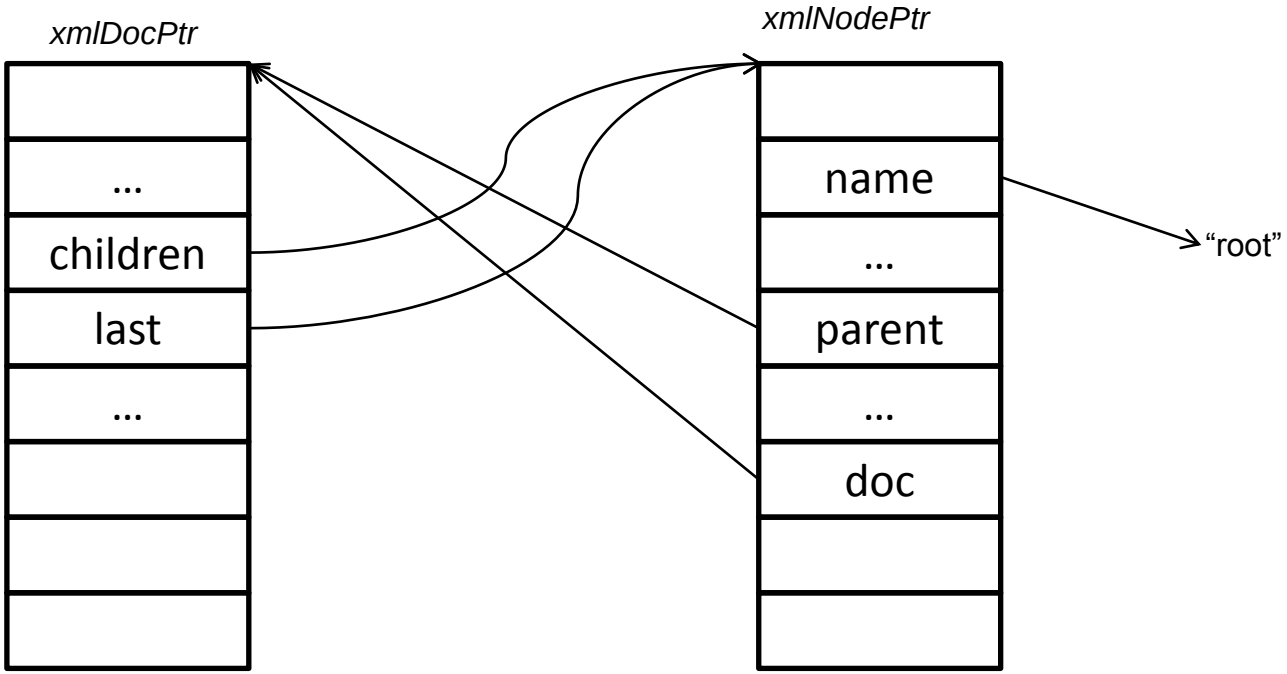
观察doc中各个成员变量的值

分析xmlNode内存结构

```
typedef struct _xmlNode xmlNode;
typedef xmlNode *xmlNodePtr;
struct _xmlNode {
    void                *_private;
    xmlElementType       type;
    const xmlChar       *name;
    struct _xmlNode     *children;
    struct _xmlNode     *last;
    struct _xmlNode     *parent;
    struct _xmlNode     *next;
    struct _xmlNode     *prev;
    struct _xmlDoc      *doc;
    xmlNs               *ns;
    xmlChar             *content;
    struct _xmlAttr     *properties;
    xmlNs               *nsDef;
    void                *psvi;
    unsigned short       line;
    unsigned short       extra;
};
```

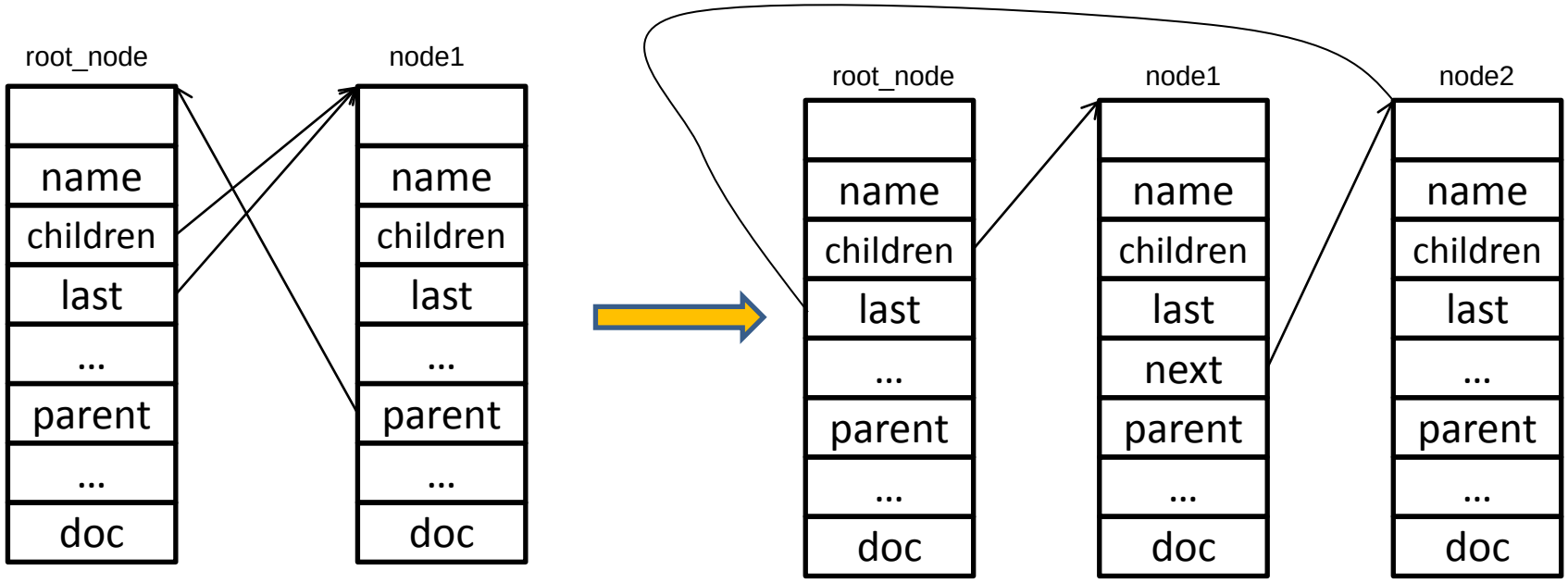
```
#include <libxml/xmlmemory.h>
#include <libxml/parser.h>
#include <stdlib.h>

int main()
{
    xmlDocPtr doc = xmlNewDoc(BAD_CAST"1.0");
    xmlNodePtr root_node = xmlNewNode(NULL, BAD_CAST"root");

    xmlDocSetRootElement(doc, root_node);

    xmlNodePtr node1 = xmlNewNode(NULL, BAD_CAST"node1");
    xmlNodePtr node2 = xmlNewNode(NULL, BAD_CAST"node2");
    xmlNodePtr node3 = xmlNewNode(NULL, BAD_CAST"node3");

    xmlAddChild(root_node, node1);
    xmlAddChild(root_node, node2);
    xmlAddChild(node2, node3);

    int nRel = xmlSaveFile("example.xml", doc);
    if (nRel != -1) {
        printf("save succeed.\n");
    }

    xmlFreeDoc(doc);
    return 1;
}
```

xmlDocSetRootElement(doc, root_node);



xmlAddChild(root_node, node1);

xmlAddChild(root_node, node2);



练习四

画出xml节点内存布局

```c
#include <libxml/xmlmemory.h>
#include <libxml/parser.h>
#include <stdlib.h>

int main()
{
    xmlDocPtr doc = xmlNewDoc(BAD_CAST"1.0");
    xmlNodePtr root_node = xmlNewNode(NULL, BAD_CAST"root");

    xmlDocSetRootElement(doc, root_node);

    xmlNewTextChild(root_node, NULL, BAD_CAST "newNode1", BAD_CAST "newNode1 content");
    xmlNewTextChild(root_node, NULL, BAD_CAST "newNode2", BAD_CAST "newNode2 content");
    xmlNewTextChild(root_node, NULL, BAD_CAST "newNode3", BAD_CAST "newNode3 content");

    xmlNodePtr node = xmlNewNode(NULL, BAD_CAST"node2");
    xmlNodePtr content = xmlNewText(BAD_CAST"NODE CONTENT");
    xmlAddChild(root_node, node);
    xmlAddChild(node, content);
    xmlNewProp(node, BAD_CAST"attribute", BAD_CAST "yes");

    node = xmlNewNode(NULL, BAD_CAST "son");
    xmlAddChild(root_node, node);
    xmlNodePtr grandson = xmlNewNode(NULL, BAD_CAST "grandson");
    xmlAddChild(node, grandson);
    xmlAddChild(grandson, xmlNewText(BAD_CAST "This is a grandson node"));

    int nRel = xmlSaveFile("CreatedXml.xml", doc);
    if (nRel != -1) {
        printf("save error.\n");
    }

    xmlFreeDoc(doc);
    return 1;
}
```

Q & A

Thank You