TypeCodes

vsnprintf字符串格式化输出实例：日志中打印程序名和行号

作者：vfhky | 时间：2016-03-13 23:28 | 分类：cseries

在Linux C/C++程序中打印日志时，可能会由于需要打印未知个数的变量参数，那么 vsnprintf 函数就排上用场了。这里使用一个简单的C程序例子，演示在打印源程序文件名和该打印函数所在的行号的同时，使用vsnprintf函数打印个数未知的参数变量。

1 完整程序

代码比较简单，如果需要把内容打印在日志文件中的话，还需要调用文件处理函数。为了便于理解，这里直接把内容输出到控制台上了。需要说明的三个地方：

```
静态全局变量 c_FileName 和 i_FileLineNum 分别用于存储源程序文件名和打印函数所在的行号；
自定义标识符 PRINT 先调用源程序文件名和行号的赋值函数 Get_File_Line ，然后调用个数未知的参数的处理
类似于sprintf和snprintf这两个函数，相比vsprintf函数，vsnprintf加了最大字节( MAXBYTES )的限制，
```

具体代码如下：

```c
/**
 * @FileName    vsnprintf_name_line.c
 * @Describe    A simple example for using vsnprintf to print the name and line-num of
 * @Author      vfhky 2016-03-13 23:28 https://typecodes.com/cseries/vsnprintffilename
 * @Compile     gcc vsnprintf_name_line.c -o vsnprintf_name_line
 */
#include <stdio.h>
#include <string.h>
#include <stdarg.h>

#define FILENAME_LEN 100
#define MAXLINE 1024
#define MAXBYTES 50

static char c_FileName[FILENAME_LEN];
static int  i_FileLineNum;

//Self-define a function which can print the name and line-number of the source file c
#define PRINT Get_File_Line( __FILE__, __LINE__ );\
              F_vsnprintf

/**
```

```
 * Get the linenum and filename of the source file.
 * @Para-in:        p_FileName: The name of the source file.
 * @Para-in:        i_FileLine: The line-number of the source file.
 */
void Get_File_Line( char *p_FileName, int i_FileLine )
{
    strcpy( c_FileName, p_FileName );
    i_FileLineNum = i_FileLine;
    return;
}

/**
 * Print the arguments according to the first argument, name as fmt.
 */
void F_vsnprintf( char *fmt, ... )
{
    char buf[MAXLINE] = {0x00};
    snprintf( buf, MAXBYTES, "[%s:%d] ", c_FileName, i_FileLineNum );
    va_list ap;
    va_start( ap, fmt );
    vsnprintf( buf+strlen(buf), MAXLINE, fmt, ap );
    va_end( ap );

    strcat( buf, "\n" );
    fflush( stdout );

    fputs( buf, stderr );
    fflush( NULL );
    return;
}

int main( int argc, char **argv )
{
    PRINT( "[%s]", "Hello." );
    PRINT( "[%s %s]", "Hello", "world." );
    return 0;
}
```

2 编译执行

使用《Linux C/C++工程中可生成ELF、动/静态库文件的通用Makefile》一文中的Makefile文件进行程序编译（当然也可以使用命令进行编译 `gcc vsnprintf_name_line.c -o vsnprintf_name_line` ），接着执行该程序，得到如下图所示的结果：

```
[vfhky@typecodes daemonize]$ make
gcc -std=c99 -D_GNU_SOURCE  -g -Wall  -I /home/vfhky/include/ -I /home/vfhky/include/C
ommAPI/ -I /home/vfhky/include/apue/ -I /home/vfhky/include/pbl/ -I /home/vfhky/includ
e/test/   -c -o vsnprintf_name_line.o vsnprintf_name_line.c
g++ -std=c99 -D_GNU_SOURCE   vsnprintf_name_line.o   -Xlinker "-("  -Xlinker "-)" -o /
home/vfhky/bin/vsnprintf_name_line
[vfhky@typecodes daemonize]$
[vfhky@typecodes daemonize]$ vsnprintf_name_line
[vsnprintf_name_line.c:56] [Hello.]
[vsnprintf_name_line.c:57] [Hello world.]
[vfhky@typecodes daemonize]$
```