

Printf本地测试说明

如果你进入到学号目录下的init文件夹的main.c文件中（如下图），你会发现这正是在运行gxemul命令后屏幕所输出的开头的内容。即如果你去修改这个文件，你就可以完成printf的本地测试了。具体流程如下：

```
int main()
{
    printf("main.c: tmain is start ...\n");
    mips_init();
    panic("main is over is error!");

    return 0;
}
```

1. 修改main.c, 添加你想要输出的内容, 如 `printf("%04d\n", 30);`。

```
printf("main.c:\tmain is start ...\n");
printf("%04d\n",30);
mips_init();
panic("main is over is error!");
```

2. `make clean`清空之前的编译内容，再使用`make`进行编译。
3. 最后运行`gxemul`，观察屏幕输出就会发现输出增加了我们希望得到的0030。

```
git@19373115:~/19373115$ /OSLAB/gxemul -E testmips -C R3000 -M 64 gxemul/vmlinux
GXemul 0.4.6    Copyright (C) 2003-2007 Anders Gavare
Read the source code and/or documentation for other Copyright messages.
```

Simple setup...

```
net: simulating 10.0.0.0/8 (max outgoing: TCP=100, UDP=100)
```

```
simulated gateway: 10.0.0.254 (60:50:40:30:20:10)
```

```
using nameserver 202.112.128.51, domain "s."
```

```
machine "default":
```

memory: 64 MB

```
cpu0: R3000 (I+D = 4+4 KB)
```

```
machine: MIPS test machine
```

```
loading gxemul/vmlinux
```

```
starting cpu0 at 0x80010000
```

```
main.c: main is start ...
```

0030

```
init.c: mips init() is called
```

```
panic at init.c:24: ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

注意：第一步修改的main.c一般会在测试时被标准文件替换，因此可以在本地放心修改