



東南大學
SOUTHEAST UNIVERSITY

东南大学本科实验报告

课程名称: _____ 你的课程

实验名称: _____ 你的实验

姓 名: _____ 你的名字

学 号: _____ 你的学号

学 院: _____ 你的学院

专 业: _____ 你的专业

2022 年 4 月 15 日

1 实验目的

本实验旨在学习如何在 Linux 系统中创建并使用多进程。学生需要在实验开始前阅读课本及相关参考资料。

2 实验内容

2.1 任务一

任务描述：

编写一个 *CPU bound* 的 C 程序和一个 *I/O bound* 的 C 程序。编译并执行这两个程序。用 *top* 命令观察这两个程序的 CPU 占用率并分析。

设计思路：

CPU bound 的 C 程序功能为重复执行乘 2 除 2 的计算，*I/O bound* 的 C 程序功能为循环读取一个 txt 文件并打印其中字符。用 *nohup* 命令将两个程序挂起在后台，再用 *top* 命令观察两个程序的 CPU 占用。

实验过程：

首先使用 `vim cpu_bound.c` 编写程序如下：

```
1 void main()
2 {
3     int a=1;
4     while(1)
5     {
6         a=a*2;
7         a=a/2;
8     }
9 }
```

再使用 `vim io_bound.c` 编写程序如下，其中读取 `str.txt` 大小为 4mb，其部分内容见附录。

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 void main()
4 {
```

```

5   while (1)
6   {
7       FILE* fp=fopen("str.txt","r");
8       int c;
9       while(c!=EOF)
10      {
11          c = fgetc(fp);
12          printf("%c",c);
13      }
14      fclose(fp);
15  }
16
17 }

```

输入以下命令编译这两个程序：

```

gcc cpu_bound.c -o cpu_bound
gcc io_bound.c -o io_bound

```

输入以下命令同时执行 `cpu_bound` 和 `io_bound`，并挂起在后台：

```

nohup ./cpu_bound &
nohup ./io_bound &

```

使用 `top` 命令观察两个程序的资源占用情况：

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
14652	root	20	0	4212	356	280	R	97.7	0.0	0:19.54	cpu_bound
14694	root	20	0	4348	512	416	R	88.4	0.0	0:10.06	io_bound

图 1: 任务一程序资源占用情况

结果分析：

从 `top` 命令我们可以看到，`cpu_bound` 比 `io_bound` 的 `cpu` 占用率高很多。这是因为 `io_bound` 有大量的 I/O 操作，导致 CPU 空闲时间比较长。而 `cpu_bound` 的工作全部是计算，所以 CPU 使用率很高。

3 总结与体会

FUC*K U SEU :)