

计算机科学与技术学院

计算机系统原理课程实验报告

实验题目：龙芯平台初探		学号：202200130048
班级：6	姓名：陈静雯	
Email: 1205037094@qq.com		
<p>实验目的：</p> <p>(1)了解数据在计算机内的编码表示，熟悉常见字符的 16 进制 ASCII 码；</p> <p>(2)初步掌握 C 语言，能够用 C 语言进行简单编程实验，掌握键盘输入及显示输出的方法；</p> <p>(3)熟悉 Linux 环境，学会如何在 Linux 环境下完成 C 程序的编写和运行。</p> <p>(4)以读写文件为例，掌握计算机系统中程序的执行流程。</p>		
<p>实验软件和硬件环境：</p> <p>多路处理器计算机教学实验系统是由四片四核龙芯 3A 处理器构成的 16 核 CC-NUMA 结构、内可配置外可扩展结构的实验硬件平台。</p>		
<p>实验原理和方法：</p> <p>(1)</p> <p>十六进制 asc 码：直接以十六进制数输出输入的字符即可，字符在计算机内部以 asc 码形式存储</p> <p>大小写转换：按位异或 32，即 0x20</p> <p>(2)</p> <p>RSA 加密解密原理，加密和解密使用不同的密钥，即使用加密密钥进行加密、解密密钥进行解密。在 RSA 算法中，加密密钥（即公开密钥）PK 是公开信息，而解密密钥（即秘密密钥）SK 是需要保密的。加密算法 E 和解密算法 D 也都是公开的。虽然解密密钥 SK 是由公开密钥 PK 决定的，由于无法计算出大数 n 的欧拉函数 $\phi(N)$，所以不能根据 PK 计算出 SK。</p> <div>$\text{密文} = \text{明文}^E \bmod N$</div> <div>$\text{公钥} = (E, N)$</div> <div>$\text{明文} = \text{密文}^D \bmod N$</div> <div>$\text{私钥} = (D, N)$</div> <p>1. 求 $n=p*q$：</p> <p>准备两个互质数 p，q。这两个数不能太小，太小容易破解。如果互质数 p 和 q 足够大，根据目前的计算机技术和其他工具，至今也没能从 N 分解出 p 和 q。</p> <p>2. $m=\text{lcm}(p-1, q-1)$</p>		

3. 求 e

$$1 < E < L$$

$$\gcd(E, L) = 1$$

4. 求 d (d 足够大)

$$1 < D < L$$

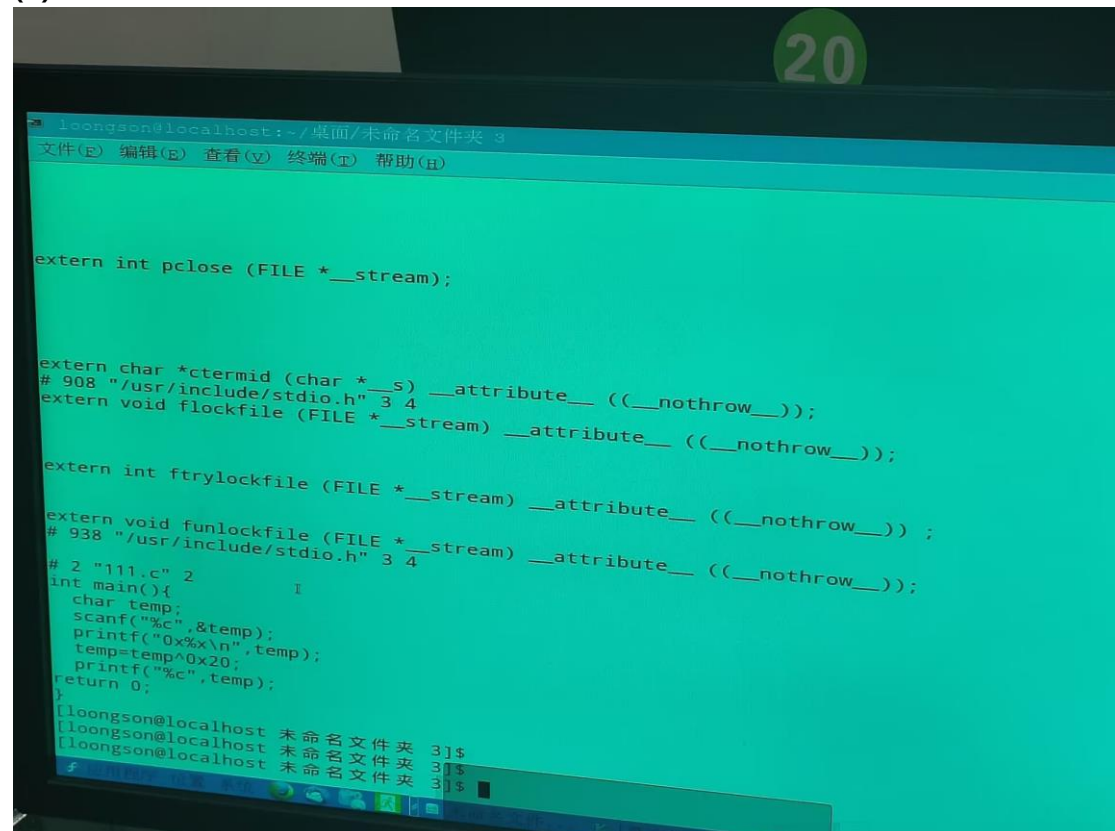
$$E * D \bmod L = 1$$

实验步骤:

实验 1.1

(1) 显示键入的字符的 16 进制 ASCII 码;

(2) 将键入的字符进行大小写转换并显示。

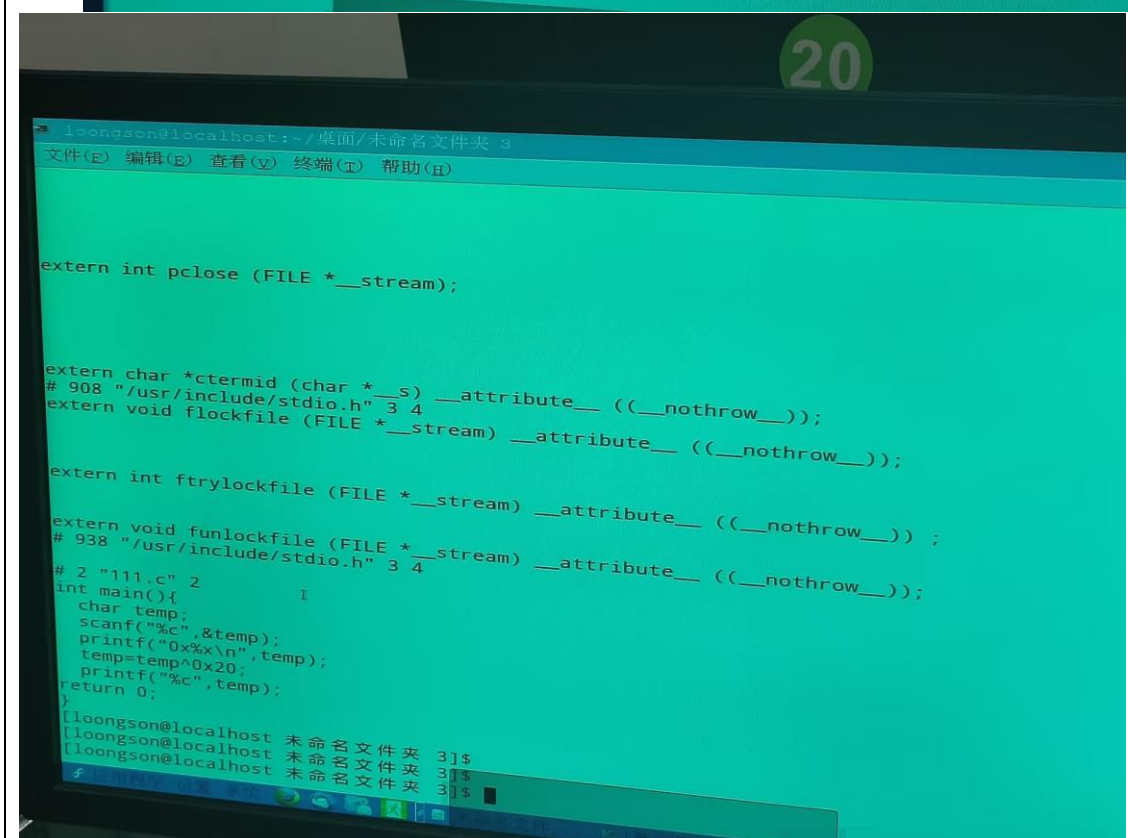
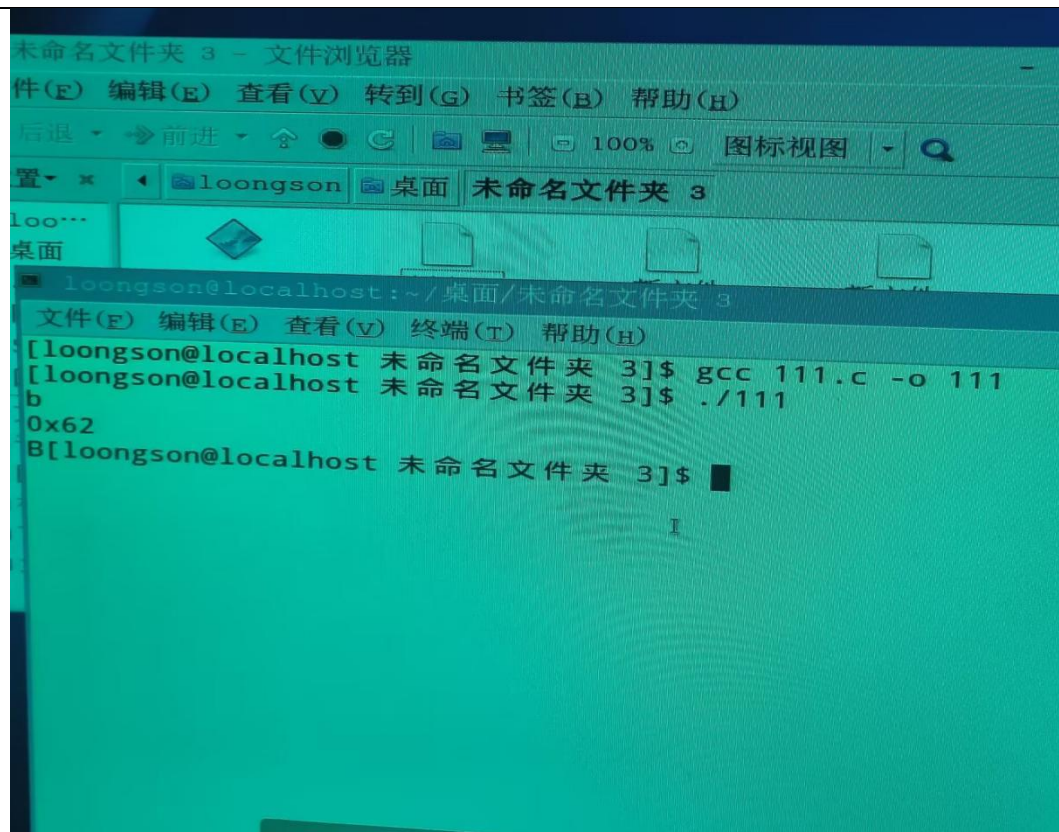


The screenshot shows a terminal window with a dark background and a green title bar. The title bar contains the text "loongson@localhost: ~/桌面/未命名文件夹 3" and a menu bar with "文件(F)", "编辑(E)", "查看(V)", "终端(T)", and "帮助(H)". The terminal displays C code for a program named "111.c". The code includes headers for `stdio.h` and `unistd.h`, and defines several external functions: `pclose`, `ctermid`, `flockfile`, `ftrylockfile`, and `funlockfile`. The `main` function takes two arguments, "111.c" and "2", and uses `scanf` to read a character into `temp`. It then prints the character, its hexadecimal value (`temp^0x20`), and the character again. The terminal output shows the program's execution, with the character '3' being input and its hexadecimal value '0x33' being printed.

```
extern int pclose (FILE *__stream);

extern char *ctermid (char *__s) __attribute__ ((__nothrow__));
# 908 "/usr/include/stdio.h" 3 4
extern void flockfile (FILE *__stream) __attribute__ ((__nothrow__));

extern int ftrylockfile (FILE *__stream) __attribute__ ((__nothrow__));
extern void funlockfile (FILE *__stream) __attribute__ ((__nothrow__));
# 938 "/usr/include/stdio.h" 3 4
# 2 "111.c" 2
int main() {
    char temp;
    scanf("%c", &temp);
    printf("%c", temp);
    temp = temp ^ 0x20;
    printf("%c", temp);
    return 0;
}
[loongson@localhost 未命名文件夹 3]$
[loongson@localhost 未命名文件夹 3]$
[loongson@localhost 未命名文件夹 3]$
```




```

/* Acquire ownership of STREAM.  */
extern void flockfile (FILE *__stream) __attribute__ ((__nothrow__

/* Try to acquire ownership of STREAM but do not block if it
   possible.  */
extern int ftrylockfile (FILE *__stream) __attribute__ ((__nothrow__

/* Relinquish the ownership granted for STREAM.  */
extern void funlockfile (FILE *__stream) __attribute__ ((__nothrow__

# 926 "/usr/include/stdio.h" 3 4
/* If we are compiling with optimizing read this file.  It contains
   several optimizing inline functions and macros.  */
# 938 "/usr/include/stdio.h" 3 4

# 2 "111.c" 2
int main(){
    char temp;
    scanf("%c",&temp);
    printf("0x%x\n",temp);
    temp=temp^0x20;
    printf("%c",temp);
    return 0;
}
[loongson@localhost 未命名文件夹] $ gcc -C 111.c -o 111.o
gcc: GCC does not support -C or -CC without -E
[loongson@localhost 未命名文件夹] $ gcc -save-temps 111.c
[loongson@localhost 未命名文件夹] $ ls
111  111.c  111.i  111.o  111.s
[loongson@localhost 未命名文件夹] $

```

20

```

loongson@localhost: ~/桌面/未命名文件夹
文件(F) 编辑(E) 查看(V) 终端(T) 帮助(H)

move    $25,$2
jalr    $25
nop

lw      $28,16($fp)
lb      $2,$24($fp)
nop
xori    $2,$2,0x20
sll     $2,$2,24
sra     $2,$2,24
sb      $2,$24($fp)
lb      $2,$24($fp)
nop
move    $4,$2
lw      $2,%call16(putchar)($28)
nop
move    $25,$2
jalr    $25
nop

lw      $28,16($fp)
move    $2,$0
move    $sp,$fp
lw      $31,36($sp)
lw      $fp,32($sp)
addiu   $sp,$sp,40
j       $31
nop

.set    macro
.set    reorder
.end    main
.size   main,.-main
.ident  "GCC" (GNU) 4.4.4
[loongson@localhost 未命名文件夹] $

```

实验 1.2

- (1)准备好要加密的文件
- (2)读入文件，进行加密操作，得到加密后的文件
- (3)对加密后的文件进行解密操作
- (4)验证解密后文件与要加密文件的一致性

设： $p = 17$, $q = 13$, (或者 $p=17$, $q=19$)

- (1) N : $N = p * q = 221$;
- (2) L : $L = \text{lcm}(p-1, q-1) = \text{lcm}(16, 12) = 48$;
- (3) E : $1 < E < L$, $\text{gcd}(E, L) = 1$, 即 $1 < E < 144$, $\text{gcd}(E, 48) = 1$, 可得 $E = 7$, 此时公钥 = $(E, N) = (7, 221)$;
- (4) 求 D : $1 < D < L$, $E * D \bmod L = 1$, 即 $1 < D < 48$, $7 * D \bmod 48 = 1$, 可得 $D=7$, 此时私钥 = $(D, N) = (7, 221)$;
- (5) 准备的明文必须是小于 N 的数，因为加密或者解密都要 $\bmod N$, 其结果必须小于 N 。

求 n, l, e, d 的代码:

```
long long p = 17, q = 13;           //互质数 p, q
//long long p = read(), q = read();读取输入的 p、q
long long n = p * q;
long long l = lcm(p - 1, q - 1); //p - 1 和 q - 1 的最小公倍数
long long e = 7;                   //e 是一个比 1 大比 m 小的数，e 和 m 的最大公
约数为 1;
if (gcd(e, l) != 1)
{
    printf("e is error\n");
    return 0;
}
long long d;
for(int i=0;i<100;i++){             //e*d mod l = 1 穷举暴力求解
    if(floor((double)(l*i+1)/(double)e)==(double)(l*i+1)/(double)e){
        d=(l*i+1)/e;
        break;
    }
}

//long long y;
//exgcd(e, l, d, y);    或辗转相除求解 d
//d = (d % l + l) % l;
```

加解密部分代码：

```
long long qpow(long long a, long long b, long long mod)
//比如 b 是 1101，即要乘 1+4+8 遍，而 a 不断平方，即 1、2、4、8 倍，b
相应的位是 1，则 ans 乘 a
```

```
{
    long long ans = 1;
    while (b)
    {
        if (b & 1) ans = ans * a % mod;    //按位与，看 b 的最后一位是否是 1
                                           //若是，乘到结果里

        a = a * a % mod;    //a 不断平方
        b >>= 1;            //b 右移一位
    }
    return ans;
}
```

```
int main()
{
    printf("input as following:\n1(ency) filename e n\n2(decry) filename d
n\n");
    int ope;
    scanf("%d", &ope);
    char s[1000];
    scanf("%s", s);

    if (ope == 1)    //加密
    {
        long long e = read(), n = read();    //e=7,n=221

        FILE *fp1 = freopen(s, "r", stdin);    //文件作为输入
        strcat(s, ".rsa");
        FILE *fp2 = freopen(s, "w", stdout);    //输出到文件
        char c = getchar();
        while (c != EOF)
```

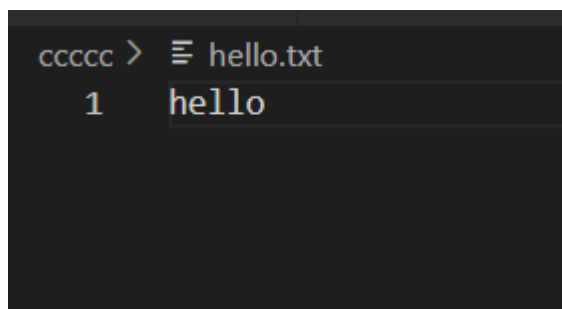
```

{
    long long x = c;
    print(qpow(x, e, n));
    putchar(' ');
    c = getchar();
}
fclose(fp1);
fclose(fp2);
}
else //解密
{
    long long d = read(), n = read(); //d=7,n=221

    FILE *fp1 = freopen(s, "r", stdin);
    s[strlen(s) - 4] = '\0';
    FILE *fp2 = freopen(s, "w", stdout);
    long long x;
    while (x = read())
        printf("%c", (char)qpow(x, d, n));
    fclose(fp1);
    fclose(fp2);
}
return 0;
}

```

原文件：



加密文件：

```
C key.c  C 111.c  hello.txt.rsa X  hello.txt
cccc > hello.txt.rsa
1 168 271 109 109 42
```

解密文件：

```
cccc > hello.txt
1 hello
```

验证解密文件与原文件一致

结论分析与体会：

问题：GCC does not support -C or -CC without -E
指定-E 编译选项，使得只输出预编译结果：结果报错

```
/* Acquire ownership of STREAM. */
extern void flockfile (FILE *__stream) __attribute__ ((__nothrow__

/* Try to acquire ownership of STREAM but do not block if it
   possible. */
extern int trylockfile (FILE *__stream) __attribute__ ((__nothrow__

/* Relinquish the ownership granted for STREAM. */
extern void funlockfile (FILE *__stream) __attribute__ ((__nothrow__
# 926 "/usr/include/stdio.h" 3 4
/* If we are compiling with optimizing read this file.  It contains
   several optimizing inline functions and macros. */
# 938 "/usr/include/stdio.h" 3 4

# 2 "111.c" 2
int main(){
    char temp;
    scanf("%c",&temp);
    printf("0x%x\n",temp);
    temp=temp^0x20;
    printf("%c",temp);
    return 0;
}
[loongson@localhost 未命名文件夹 3]$ gcc -C 111.c -o 111.o
gcc: GCC does not support -C or -CC without -E
[loongson@localhost 未命名文件夹 3]$ gcc -save-temps 111.c
111 111.c 111.i 111.o 111.s
[loongson@localhost 未命名文件夹 3]$ ls
111.c 111.i 111.o 111.s 111.out 新文件 新文件.c
```