中国科学技术大学

一九九八年招收硕士学位研究生入学考试试题
试题名称:编译原理与操作系统

一. （10 分）某操作系统下合法的文件名为

device:name.extension

其中第一部分（device:）和第三部分（.extension）可省略,若 device,name 和 extension 都是字母串,长度不限,但至少为 1,画出识别这种文件名的确定有限自动机.

二. （10 分）下面文法是否为 LL(1)文法? 说明理由.

S ---> A B | P Q x　　A ---> x y　　B ---> b c

P ---> d P | **ε**　　Q ---> a Q | **ε**

三. （10 分）把表达式

-(a + b) * (c + d) + (a + b + c)

翻译成四元式.

四. （10 分）UNIX 下的 C 编译命令 cc 的选择项 g 和 O 的解释如下,其中 dbx 的解释是“dbx is an utility for source-level debugging and excution of programs written in C ”.试说明为什么用了选择项 g 后,选择项 O 遍被忽略.

```
-g          Produce additional symbol table information for
            dbx(1) and dbxtool(1) and pass -lg option to ld(1)
            (so as to include the g library, that is :
            /usr/lib/libg.a).  When this option is given, the
            -O and -R options are suppressed.
-O[level]   Optimize the object code. Ignored when either -g, -go,
            or -a is used.    …
```

五. （10 分）下面程序在 SUN 工作站上运行时陷入死循环,试说明原因.如果将第 6 行的 long *p 改成 short *p, 并且将第 16 行 long k 改成 short k 后,loop 中的循环体执行一次便停止了.试说明原因.

```
Main()
{
   addr();
   loop();
}
long *p;  /* 第 6 行 */
loop()
{  long  i,j;
   j=0;
   for(i=0;i<10;i++)
   {   (*p) - -;
       j++;
   }
}
```

```
addr()
{   long k;    /*  第 16 行 */
    k=0;
    p=&k;
}
```

六. 填空(每空 1 分,共 20 分)

1. 用户与操作系统之间的接口主要分为_____和_____两类.
2. 在操作系统中,不确定性主要是指_____和_____.
3. 在 UNIX 系统 V 中,新建的子进程从父进程那里继承了_____、____和___等多种资源.
4. 在可变分区存储管理中,分区的保护通常采用_____和_____两种方式.
5. 逻辑设备表(LUT)的主要功能是_____和_____.
6. 在采用请求分页式存储管理的系统中,地址变换过程可能会因为_____、_____和_____等原因而产生中断.
7. 在UNIX系统V中,如果一个盘块的大小为1KB,每个盘号占4个字节,那么,一个进程要访问偏移量为263168字节处的数据时,需要经过___次间址.
8. 设备驱动程序是一种低级的系统例程,它通常分为____和_____两部分.
9. UNIX 系统 V 在打开(open)一个文件时,需要为其分配_____,_____和_____等多种资源.

七. (10 分)简述 LRU,NRU 和 LFU 三种页面置换算法的思想,并各给出一种可能的实现方案.

八. (10 分)何谓临界区?下面给出的实现两个进程互斥的算法是安全的吗?为什么?

```
#define  TRUE   0
#define  FALSE  1
int flag[2];
flag[0] = flag[1] = FALSE;
enter_crtsec(i)
int i;
{   while(flag[1-i]);
    flag[i]=TRUE;
}
leave_crtsec(i)
int i;
{   flag[i]=FALSE;
}
process i:
          …
          enter_crtsec(i); /* 进入临界区 */
              IN CRITICAL SECTION
          leave_crtsec(i); /* 离开临界区 */
          …
```

九. (10 分)要使一个系统不发生死锁,一般可采用那些方法,简述它们的实现原理.