作业3

1．若在后备作业队列中等待运行的同时有三个作业1、2、3，已知它们各自的运行时间为a、b、c，且满足关系a<b<c，试证明采用短作业优先调度算法能获得最小平均周转时间。

解：采用短作业优先调度算法，总周转时间为：

T1=a+(a+b)+(a+b+c)=3a+2b+c (1)

若不按短作业优先调度算法来调度这三个作业，不失一般性，假设调度顺序为2、1、3，则其总周转时间为：

T2=b+(b+a)+(b+a+c)=3b+2a+c (2)

(2)-(1)式，得：

T2-T1=b-a>0

可见，短作业优先调度算法能获得最小平均周转时间。

2．设有4道作业，它们的提交时间及执行时间如下：

|  |  |  |
| --- | --- | --- |
| 作业号 | 提交时间 | 执行时间 |
| 1  2  3  4 | 10.0  10.2  10.4  10.5 | 2.0  1.0  0.5  0.3 |

试计算在单道程序环境下，采用先来先服务调度算法、最短作业优先调度算法、响应比优先调度算法时的平均周转时间和平均带权周转时间，并指出它们的调度顺序。（时间单位：小时，以十进制进行计算）

解：若采用先来先服务调度算法，调度顺序为1、2、3、4

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 作业号 | 提交时间 | 执行时间 | 开始时间 | 完成时间 | 周转时间 | 带权周转时间 |
| 1  2  3  4 | 10.0  10.2  10.4  10.5 | 2.0  1.0  0.5  0.3 | 10.0  12.0  13.0  13.5 | 12.0  13.0  13.5  13.8 | 2.0  2.8  3.1  3.3 | 1.0  2.8  6.2  11.0 |

平均周转时间：T=(2.0+2.8+3.1+3.3)/4=2.8

平均带权周转时间：W=(1+2.8+6.2+11.0)/4=5.25

若采用最短作业优先服务调度算法，调度顺序为1、4、3、2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 作业号 | 提交时间 | 执行时间 | 开始时间 | 完成时间 | 周转时间 | 带权周转时间 |
| 1  4  3  2 | 10.0  10.5  10.4  10.2 | 2.0  0.3  0.5  1.0 | 10.0  12.0  12.3  12.8 | 12.0  12.3  12.8  13.8 | 2.0  1.8  2.4  3.6 | 1.0  6.0  4.8  3.6 |

平均周转时间：T=(2.0+1.8+2.4+3.6)/4=2.45

平均带权周转时间：W=(1+6+4.8+3.6)/4=3.85

若采用响应比优先服务调度算法，调度顺序为1、4、3、2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 作业号 | 提交时间 | 执行时间 | 开始时间 | 完成时间 | 周转时间 | 带权周转时间 |
| 1  4  3  2 | 10.0  10.5  10.4  10.2 | 2.0  0.3  0.5  1.0 | 10.0  12.0  12.3  12.8 | 12.0  12.3  12.8  13.8 | 2.0  1.8  2.4  3.6 | 1.0  6  4.8  3.6 |

J1结束后，计算响应比

R2=1+(12-10.2)/1=2.8

R3=1+(12-10.4)/0.5=4.2

R4=1+(12-10.5)/0.3=6 J4被调度

J4结束后，计算响应比

R2=1+(12.3-10.2)/1=3.1

R3=1+(12.3-10.4)/0.5=4.8 J3被调度

平均周转时间：T=(2.0+1.8+2.4+3.6)/4=2.45

平均带权周转时间：W=(1+6+4.8+3.6)/4=3.85

作业4

2.有一请求分页存储管理系统，页面大小为每页100字节，有一个50\*50的整型数组按行连续存放，每个证书占两个字节，将数组初始化为0的程序描述如下：

int a[50][50];

int i,j;

for(i=0;i<=49;i++)

for(j=0;j<=49;j++)

a[i][j]=0;

若在程序执行时内存中只有一个存储块用来存放数组信息，试问该程序执行时产生多少次缺页中断？若交换内外循环的次序，缺页中断次数为多少？

解：由题可知，数组共有2500个整数，每个整数占2字节，共需存储空间5000字节，而页面大小为100字节（正好存放50个整数），故数组占用逻辑空间50页，假设数据从该作业的第m页开始存放，则数组分布在第m页到m+49页中，在主存中的排列顺序为：

a[0][0]，a[0][1]，…，a[0][49]

a[1][0]，a[1][1]，…，a[1][49]

…

a[49][0]，a[49][1]，…，a[49][49]

由于该初始化程序是按行进行的，因此每次缺页中断调进1页后，可以读取50个整数，然后再调入下一页，故缺页次数为50次。

若交换内外循环的次序，每次调页只能读取一个数组元素，故缺页中断次数为2500次。

3. 设有一页式存储管理系统，向用户提供的逻辑地址空间最大为16页，每页2048字节，内存总共有8个存储块，试问逻辑地址至少应为多少位？内存空间有多大？

解：页式存储管理中的逻辑地址为：

页号+页内位移量

本题中，每页2048字节，所以需要(2040)10=(100000000000)2，=211，表达0~2047至少需要11个二进制位，逻辑地址空间最大为16页，页号需要4个二进制位，故逻辑地址至少应为15位。

4.若在一分页存储管理系统中，某作业的页表如下所示。已知页面大小为1024字节，试将逻辑地址1011、2148、3000、4000、5012转化为相应的物理地址。

|  |  |
| --- | --- |
| 页号 | 块号 |
| 0  1  2  3 | 2  3  1  6 |

解：设页号为P，页内位移为W，逻辑地址为A，页面大小为L

P=int(A/L)，W=A mod L

|  |  |  |  |
| --- | --- | --- | --- |
| 逻辑地址 | 页号 | 位移量 | 物理地址 |
| 1011 | 0 | 1011 | 2\*1024+1011=3059 |
| 2148 | 2 | 100 | 1\*1024+100=1124 |
| 3000 | 2 | 952 | 1\*1024+952=1976 |
| 4000 | 3 | 928 | 6\*1024+928=7072 |
| 5012 | 4 | 916 | 越界，逻辑地址非法 |

作业11

编写程序，并运行之，给出运行结果：

1.编写一个顺序结构的shell程序hw-1：显示当前登录的所有用户；检测某用户是否在线（如user01,wenyuan）,若在线显示0，不在线显示1(该用户名可以通过位置参数提供)；显示位置参数的个数。（此程序无需判断参数的合法性）

#hw-1

who

who|grep "^$1"

echo $?

echo $#

chmod +x hw-1

./hw-1 wenyuan

./hw-1 user01

或

#hw-1

echo "当前登陆的所有用户有:"

who|cut -f1 -d' '

echo "下面检测位置参数中的用户名是否登陆（登陆显示 0，未登陆显示 1）"

user="$1"

who|grep "^$user">/dev/null && echo $1" 0" ||echo $1" 1"

echo "位置参数一共有:"$#"个

2. 编写一个分支结构的shell程序hw-2：从键盘上输入一个脚本文件名，若不存在，显示“\*\*文件不存在”的信息，并退出程序，若存在且有执行权限直接运行，若没有执行权限，修改权限后再运行(假设该脚本不需要位置参数即可执行)。

答：

#hw-2

read –p “input filename:” filename

if [ ! -f $filename ]

then

echo "$filename does not exist"

exit 0

else

if [ ! -x $filename ]

then

chmod +x $filename

fi

./$filename

fi

chmod +x my\_sh2

$ ./hw-2

input filename:（略）

3. 编写一个菜单式shell程序hw-3，菜单内容如下：

1：chmod

2: cat

3: rename

q: exit menu

菜单功能如下：

1）输入文件名，判断文件是否存在，若存在，增加执行权限，以长列表格式显示该文件属性；如果不存在，显示“\*\*\* does not exist”并退出。

2）输入文件名，存在且有读权限，显示之；不存在，创建一个空文件。

3）输入源文件名和目标文件名，若原文件不存在，显示“Source file does not exist”；若原文件存在，即使目标文件存在也覆盖已有文件。

q）显示“quit”并退出。

其它）显示“input error”并退出。

当执行菜单1-3并正常结束，显示“The End”

答：

clear

echo "1: chmod"

echo "2: cat"

echo "3: rename"

echo "q: exit menu"

echo "plese input choose the menu(1/2/3/q):"

read choice

case $choice in

1) read -p "input a filename:" filename

if [ -f $filename ]

then

chmod +x $filename

ls -l $filename

else

echo "$filename does not exist" ;exit 0

fi

;;

2) read -p "input a filename:" filename

if [ -f $filename ] &&[ -r $filename ]

then

cat $filename

else

touch $ficlename

fi

;;

3)echo "input a source filename:"

read filename1

echo "input a object filename:"

read filename2

if [ ! -f $filename1 ]

then

echo "Source file does not exist"

exit 0

else

mv -f $filename1 $filename2

fi

;;

q)

echo "quit";exit 1

;;

\*)

echo “input error”;exit 2

;;

esac

echo "The End"

chmod +x hw-3

./hw-3

注：学生书写的菜单程序

cat << EOF

菜单

---------------------

1:chmod

2:cat

3:rename

4:exit menu

---------------------

EOF

作业12

1. 修改作业10中的习题6（hw3）为hw4，循环选择菜单:

1: chmod

2: cat

3: rename

q: exit menu

（1）选择1-3完成菜单相应的功能，选择q退出，选择其它，报错

（2）完成一次操作后，提示是否继续（y/n），若选择y或Y可重新选择菜单。

答：

while :

do

clear

echo "1: chmod"

echo "2: cat or vi"

echo “3: rename”

echo "q: exit menu"

echo "plese input a number from 1 to 4 to choose the menu:"

read choice

case $choice in

1)echo "input a filename:"

read filename

if [ ! -e $filename ]

then

echo "$filename does not exist"

exit 0

else

if [ -d $filename ]

then

chmod +x $filename

fi

if [ -f $filename ]

then

chmod +r $filename

fi

fi

;;

2)echo "input a filename:"

read filename

if [ -f $filename ]

then

cat $filename

else

vi $ficlename

fi

;;

3)echo "input a source filename:"

read filename1

echo "input a object filename:"

read filename2

if [ ! -f $filename1 ]

then

echo "Source file does not exist"

exit 0

fi

if [ -f $filename2 ]

then

mv -i $filename1 $filename2

fi

;;

q)

exit 1

;;

\*)

echo "input error"

;;

esac

read -p "是否继续（y/n）？" aa

if [ "$aa" = "Y" ]|| [ "$aa" = "y" ]

then

continue

else

exit

fi

done

2．循环测试用户名与密码是否正确，最多测试 3 次,输入正确提示登录成功,否则提示登录失败（用户名为 tom ，密码为 123456）（要求：从键盘输入用户名和密码，然后判断是否与用户名 tom ，密码 123456一致）

答：

for ((i=1;i<=3;i++))

或 for i in 1 2 3

或i=1；while [ $i -le 3 ] do … i=$(($i+1));…done 或((c = c + 1))也可以c增1

或for i in `seq 3`

do

read -p "请输入用户名:" user

read -p "请输入密码:" pass

if [ "$user" == 'tom' -a "$pass" == '123456' ];then

echo "Login successful"

exit

fi

done

echo "Login Failed"

作业13

1．编写多个源文件（speak.h、speak.cpp、c-file1.cpp），生成可执行程序，并运行之。其中：

speak.h文件：定义Speak类，类中只有一个无参数公有函数sayHello()的声明

speak.cpp文件：定义Speak类中的函数sayHello()的函数体，完成输出”Hello”

c-file1.cpp文件：定义main()，使用Speak类输出”Hello”

（1）编写三个文件的源代码

（2）编译、连接后，生成可执行文件c-file1，写出相应指令。如出现编译错误，请自行调试程序，直至无误。

（3）运行c-file1，写出相应指令和运行结果。

答：

/\* speak.h \*/

#include <iostream>

class Speak

{

public:

void sayHello();

};

/\* speak.cpp \*/

#include "speak.h"

void Speak::sayHello()

{

std::cout << "Hello \n";

}

/\* c-file1.cpp \*/

#include "speak.h"

int main()

{

Speak speak;

speak.sayHello();

}

g++ c-file1.cpp speak.cpp -o c-file1

./ c-file1

Hello

2．编写C++程序c-file2.cpp和fun1.cpp，文件中分别包含两个函数main()和isprime()。运行程序时，从键盘输入一个整数，判断其是否为素数，输出不同信息。

（1）编辑c-file2.cpp和fun1.cpp两个源文件。

（2）分别编译两个文件，直到没有错误再连接、运行程序（可执行文件名为c-file2）。如果有错误，请调试程序直至无误。

（3）运行该程序

vi my\_c2.cpp

//c-file2.cpp

#include <iostream>

using namespace std;

int isprimer(int );

int main()

{int x;

cin>>x;

if(isprimer(x))

cout<<"is primer!"<<endl;

else

cout<<"is not primer!"<<endl;

}

vi fun1.cpp

//fun1.cpp

#include <math.h>

int isprimer(int n)

{

int sqrtm=sqrt(n);

for(int i=2; i<=sqrtm; i++)

if(n%i==0)

return 0;

return 1;

}

g++ -c c-file2.cpp

g++ -c fun1.cpp

g++ c-file2.cpp fun1.cpp –o c-file2

./c-file2

3. 复数的加法运算

方法一：类的成员函数实现（要求含有两个重载的构造函数，无参、有参），c-file3.cpp（也可以写在多个源文件中，最后可执行文件名为c-file2）

（1）定义complex类，含有实部和虚部两个变量，两个构造函数（执行构造函数时给出必要的输出信息，如：“构造函数1被执行”，“构造函数2被执行”等），成员函数Add()完成两个复数加法运算，成员函数Disp()完成复数的输出。

（2）主函数中定义两个复数类型的对象，完成复数加法，并输出a+bi格式的累加结果。

#include<iostream>

using namespace std;

class complex

{

float a,b;

public:

complex(){a=0.0;b=0.0; cout<<"默认构造函数1执行！"<<endl;}

complex(float x,float y) {a=x;b=y; cout<<"默认构造函数2执行！"<<endl;}

void Add(complex& comp1,complex& comp2);

void disp()

{

if(b>0)

cout<<a<<"+"<<b<<"i"<<endl;

else

cout<<a<<"-"<<-b<<"i"<<endl; //避免3+-2i形式出现

}

};

void complex::Add(complex& comp1,complex& comp2)

{ float x=comp1.a+comp2.a;

float y=comp1.b+comp2.b;

cout<<"复数s1、s2之和为：";

if(y>0)

cout<<x<<"+"<<y<<"i"<<endl;

else

cout<<x<<"-"<<-y<<"i"<<endl; //避免3+-2i形式出现

}

int main()

{

complex s1(1,2),s2(3,4),s3;

cout<<"复数s1:";

s1.disp();

cout<<"复数s2:";

s2.disp();

s3.Add(s1,s2);

cout<<endl;

}

方法二：修改c-file2.cpp，用友元函数实现同样功能c-file3.cpp

#include<iostream>

using namespace std;

class complex

{

float a,b;

public:

complex(){a=0.0;b=0.0;cout<<"默认构造函数1执行！"<<endl;}

complex(float x,float y) {a=x;b=y;cout<<"构造函数2执行！"<<endl;}

friend void Add(complex& comp1,complex& comp2);

void disp()

{

if(b>0)

cout<<a<<"+"<<b<<"i"<<endl;

else

cout<<a<<"-"<<-b<<"i"<<endl;

}

};

void Add(complex& comp1,complex& comp2)

{

float x=comp1.a+comp2.a;

float y=comp1.b+comp2.b;

if(y>0)

cout<<x<<"+"<<y<<"i"<<endl;

else

cout<<x<<"-"<<-y<<"i"<<endl;

}

int main()

{

complex s1(1,2),s2(3,4);

cout<<"复数s1:";

s1.disp();

cout<<"复数s2:";

s2.disp();

cout<<"复数s1、s2之和为：";

Add(s1,s2);

cout<<endl;

}

方法三：运算符重载实现c-file4.cpp

#include<iostream>

using namespace std;

class complex

{

float a,b;

public:

complex(){a=0.0;b=0.0; cout<<"默认构造函数1执行！"<<endl;}

complex(float x,float y) {a=x;b=y; cout<<"默认构造函数2执行！"<<endl;}

void operator+(complex& comp1)

{ float x=a+comp1.a;

float y=b+comp1.b;

cout<<"复数s1、s2之和为：";

if(y>0)

cout<<x<<"+"<<y<<"i"<<endl;

else

cout<<x<<"-"<<-y<<"i"<<endl; //避免3+-2i形式出现

}

void disp()

{ if(b>0)

cout<<a<<"+"<<b<<"i"<<endl;

else

cout<<a<<"-"<<-b<<"i"<<endl; //避免3+-2i形式出现

}

};

int main()

{

complex s1(1,2),s2(3,4),s3;

cout<<"复数s1:";

s1.disp();

cout<<"复数s2:";

s2.disp();

s1+s2;

cout<<endl;

}

作业14

C++运算符重载

1．复数的加法运算（重载运算符，实现两个复数相加），用成员函数实现重载c-file5.cpp

#include<iostream>

using namespace std;

class complex

{

float a,b;

public:

complex(){a=0.0;b=0.0;}

complex(float x,float y) {a=x;b=y;}

void operator+(complex& comp1)

{ float x=a+comp1.a;

float y=b+comp1.b;

cout<<"复数s1、s2之和为：";

if(y>0)

cout<<x<<"+"<<y<<"i"<<endl;

else

cout<<x<<"-"<<-y<<"i"<<endl; //避免3+-2i形式出现

}

void disp()

{ if(b>0)

cout<<a<<"+"<<b<<"i"<<endl;

else

cout<<a<<"-"<<-b<<"i"<<endl; //避免3+-2i形式出现

}

};

int main()

{

complex s1(1,2),s2(3,4),s3;

cout<<"复数s1:";

s1.disp();

cout<<"复数s2:";

s2.disp();

s1+s2;

cout<<endl;

}

2．复数的加法运算（重载运算符，实现两个复数相加），用友元函数实现重载c-file6.cpp

#include<iostream>

using namespace std;

class complex

{

float a,b;

public:

complex(){a=0.0;b=0.0;}

complex(float x,float y) {a=x;b=y;}

friend void operator+(complex& comp1, complex& comp2)

{ float x= comp1.a+ comp2.a;

float y= comp1.b+ comp2.b;

cout<<"复数s1、s2之和为：";

if(y>0)

cout<<x<<"+"<<y<<"i"<<endl;

else

cout<<x<<"-"<<-y<<"i"<<endl; //避免3+-2i形式出现

}

void disp()

{ if(b>0)

cout<<a<<"+"<<b<<"i"<<endl;

else

cout<<a<<"-"<<-b<<"i"<<endl; //避免3+-2i形式出现

}

};

int main()

{

complex s1(1,2),s2(3,4),s3;

cout<<"复数s1:";

s1.disp();

cout<<"复数s2:";

s2.disp();

s1+s2;

cout<<endl;

}

作业15

1. 计算机little-sister.vu.nl的IP地址是130.37.62.23，那么该计算机是在A类、B类还是C类网上？写出该IP地址的网络地址、广播地址、子网掩码。

答：B类网络

网络地址：130.37.0.0

广播地址：130.37.255.255

子网掩码：255.255.0.0

2.若已知某网络地址为202.113.29.64/26，写出该网络的网络地址、广播地址、子网掩码，并判断202.113.29.68 与202.113.29.129是否为该子网内的主机？为什么？

2. 202.113.29.64/26，即11001010 01110001 00011101 01000000

网络地址：202.113.29.64，即11001010 01110001 00011101 01000000

广播地址：202.113.29.127，即11001010 01110001 00011101 01111111

子网掩码：255.255.255.192，即11111111 11111111 11111111 11000000

主机202.113.29.68，即11001010 01110001 00011101 01000100，是该网内的主机，因为

（202.113.29.68）&（255.255.255.192）=

(11001010011100010001110101000100)&(11111111111111111111111111000000)

=(11001010 01110001 00011101 01000000)=(202.113.29.64)与网络地址一致。

主机202.113.29.129，即11001010 01110001 00011101 10000001，不是该网内的主机，因为

（202.113.29.129）&（255.255.255.192）=

(11001010011100010001110110000001)&(11111111111111111111111111000000)

=(11001010 01110001 00011101 10000000)=(202.113.29.128)与网络地址不符。

作业16

二．编程：

编写一个C++程序myadd.cpp，要求如下：

（1）从键盘输入十进制点分格式IP地址和端口号（运行时从键盘输入：192.168.1.2及8080）（注：为了便于评判结果，请输入这两个值）

（2）定义struct sockaddr\_in类型的对象，对其进行地址填充

（3）输出下列三组信息：

键盘输入的IP地址、端口号

转换为网络字节顺序的IP地址、端口号

再次将网络字节顺序的IP地址和端口号转换为本机字节顺序的IP地址、端口号

（4）编译并运行，写出运行结果

答：

#include <sys/socket.h>

#include <sys/types.h>

#include <iostream>

using namespace std;

#include <stdlib.h>

#include <string.h>

#include <arpa/inet.h>

int main()

{struct sockaddr\_in saddr;

char ipaddr[16];

unsigned short port;

cout<<"input Ipaddr：";

cin>>ipaddr;

cout<<"input port：";

cin>>port;

cout<<"original:"<<ipaddr<<","<<port<<endl;

memset(&saddr,0,sizeof(struct sockaddr\_in));

saddr.sin\_family=AF\_INET;

saddr.sin\_port=htons(port);

saddr.sin\_addr.s\_addr=inet\_addr(ipaddr);

//inet\_aton(argv[1],&saddr.sin\_addr);

cout<<"convert:"<<saddr.sin\_addr.s\_addr<<","<<saddr.sin\_port<<endl;

cout<<"once more:"<<inet\_ntoa(saddr.sin\_addr)<<","<<ntohs(saddr.sin\_port)<<endl;

}