

Operating System

MP1: System Call

Team 41 分工

PartII-1 SC_Halt	郭逸洪
PartII-1 SC_Create	郭逸洪
PartII-1 SC_PrintInt	黃啓恆
PartII-1 Makefile	郭逸洪
PartII-2 Open	郭逸洪&黃啓恆
PartII-2 Write	郭逸洪&黃啓恆
PartII-2 Read	郭逸洪&黃啓恆
PartII-2 Close	郭逸洪&黃啓恆

Contents

Part II-1 Trace Code	3
(a) SC Halt	3
(b) SC Create	4
(c) SC PrintInt	5
(d) Makefile (code/test/Makefile)	9
Part II-2 I/O System Calls Implementation	11
(a) OpenFileId Open(char *name)	11
(b) int Write(char *buffer, int size, OpenFileId id)	12
(c) int Read(char *buffer, int size, OpenFileId id)	13
(d) int Close(OpenFileId id)	13
Difficulties encountered	15
Feedback	16

Part II-1 Trace Code

(a) SC_Halt

(i) machine/mipssim.cc

```
void Machine::Run()
```

- 切換到UserMode
- 在無窮迴圈中呼叫Machine::OneInstruction，在執行完指令後若未exit，呼叫Interrupt::OneTick()使kernel的時間前進SystemTick（10）或UserTick（1），取決於當下是在UserMode或SystemMode

(ii) machine/mipssim.cc

```
void Machine::OneInstruction(Instruction *instr)
```

- 在program counter紀錄的記憶體位址讀取4 Bytes的指令，若失敗則直接返回，否則繼續執行
- 執行指令解碼
- 先將PC+4但不寫回，避免執行時發現是分支指令或有錯誤發生
- 根據指令的opCode找到對應的case，halt程式編譯後產生的組合語言如下，程式主要任務來自最後兩行，將r2暫存器設置為0後，呼叫SYSCALL，進到Machine::RaiseException

```
At PC = 0      JAL 68
At PC = 4      SLL r0,r0,0
At PC = 272    ADDIU r29,r29,-24
At PC = 276    SW r31,20(r29)
At PC = 280    SW r30,16(r29)
At PC = 284    ADDU r30,r29,r0
At PC = 288    JAL 65
At PC = 292    SLL r0,r0,0
At PC = 260    JR r0,r31
At PC = 264    SLL r0,r0,0
At PC = 296    JAL 5
At PC = 300    SLL r0,r0,0
At PC = 20     ADDIU r2,r0,0
At PC = 24     SYSCALL
```

(iii) machine/machine.cc

```
void Machine::RaiseException(ExceptionType which, int badVAddr)
```

- SYSCALL將BadVAddrReg暫存器的值設為0
- 切換到SystemMode
- 呼叫ExceptionHandler

- (iv) userprog/exception.cc
 - void ExceptionHandler(ExceptionType which)
 - 將r2暫存器的值（目前為0）放到type變數
 - 進入which為SyscallException且type為SC_Halt（值為0）的case
 - 呼叫SysHalt
- (v) userprog/ksyscall.h
 - void SysHalt()
 - 呼叫kernel interrupt的Halt方法
- (vi) machine/interrupt.cc
 - void Interrupt::Halt()
 - 印出停止訊息及kernel狀態統計
 - 刪除kernel
 - 在kernel解構式中刪除各項物件、資源
 - exit 0

(b) SC_Create

- (i) userprog/exception.cc
 - void ExceptionHandler(ExceptionType which)
 - createFile程式主要由以下組合語言執行檔案創建，呼叫SYSCALL時r2暫存器值為4、r4暫存器值為400，故ExceptionHandler中的type變數為4、val變數為400

```

At PC = 296      LUI  r4,0
At PC = 300      ADDIU r4,r4,400
At PC = 304      JAL  37
At PC = 308      SLL  r0,r0,0
At PC = 148      ADDIU r2,r0,4
At PC = 152      SYSCALL

```

- 進入which為SyscallException且type為SC_Create（值為4）的case
 - 從記憶體位置為400處讀取檔案名稱
 - 呼叫SysCreate並傳入檔案名稱，回傳是否成功的狀態
 - SysCreate結束後，將狀態值寫回r2暫存器，此值將影響印出訊息的分支
 - 更新PC並返回
- (ii) userprog/ksyscall.h
 - int SysCreate(char *filename)
 - 呼叫kernel fileSystem的Create方法

(iii) filesystems/filesys.h

bool Create(char *name)

- 呼叫sysdep.cc的OpenForWrite
- 若OpenForWrite回傳的file descriptor為-1則回傳失敗
- 否則呼叫sysdep.cc的Close關閉該fd
- 回傳成功

(iv) lib/sysdep.cc

int OpenForWrite(char *name)

- 呼叫standard libraries的open在host UNIX的檔案系統上打開檔案，若檔案不存在則創建，若存在則截切原文件
- 回傳file descriptor

(v) lib/sysdep.cc

int Close(int fd)

- 呼叫standard libraries的close關閉fd

(c) SC_PrintInt

(i) userprog/exception.cc

void ExceptionHandler(ExceptionType which)

- add程式主要由以下組合語言執行PrintInt，呼叫SYSCALL時r2暫存器值為16、r4暫存器為要印出的值，故ExceptionHandler中的type變數為16、val變數為先前add的計算結果65

```
At PC = 76      JR r0,r31
At PC = 80      SLL r0,r0,0
At PC = 312     SW r2,16(r30)
At PC = 316     LW r4,16(r30)
At PC = 320     JAL 9
At PC = 324     SLL r0,r0,0
At PC = 36      ADDIU r2,r0,16
At PC = 40      SYSCALL
```

- 進入which為SyscallException且type為SC_PrintInt（值為16）的case
- 將r4暫存器的值放到val中並將其傳入SysPrintInt()中

(ii) userprog/ksyscall.h

void SysPrintInt(int val)

- 呼叫SynchConsoleOutput::PutInt，並傳入val

(iii) userprog/synchconsole.cc

void SynchConsoleOutput::PutInt(int value)

- 為了簡化實作，這邊在印出前先將int轉換為字串（字元陣列），並加上換行符號，此時字串實質內容為"65\n"
- 將SynchConsoleOutput的lock鎖上，同時間只允許一個執行緒持有該鎖，避免output錯誤，其餘執行緒只能停在此處等待該鎖的執行緒釋放該鎖
- 接下來在迴圈中呼叫ConsoleOutput::PutChar直到印完要輸出的字元陣列，由以下add程式輸出的除錯訊息可知，在同一個時刻下（此除錯訊息是在totalTicks為73時），Nachos模擬的IO裝置在ConsoleOutput::PutChar中schedule一個100 ticks後要執行的interrupt，且程式呼叫SynchConsoleOutput::waitFor的P()方法，禁用interrupt並將add這個執行緒轉為sleep狀態，直到100 ticks後，由模擬IO裝置觸發的callback將其喚醒

```
In SynchConsoleOutput::PutChar, into
consoleOutput->PutChar, 73
6Scheduling interrupt handler the console write at time
= 173
In SynchConsoleOutput::PutChar, return from
consoleOutput->PutChar, 73
In SynchConsoleOutput::PutChar, into waitFor->P(), 73
In Semaphore::P(), 73
    interrupts: on -> off
Sleeping thread: add
```

- 100 ticks後（此除錯訊息在totalTicks為173時）SynchConsoleOutput::waitFor的V()因callback而被呼叫，Nachos將add的執行緒放到ready list，之後被CPU執行，成功輸出一個字元，並重新開啟interrupt

```
Invoking interrupt handler for the
console write at time 173
In Interrupt::CheckIfDue, into
callOnInterrupt->CallBack, 173
In ConsoleOutput::CallBack(), 173
In SynchConsoleOutput::CallBack(), 173
In Semaphore::V(), 173
    interrupts: off -> off
Putting thread on ready list: add
    interrupts: off -> off
In Interrupt::CheckIfDue, return from
callOnInterrupt->CallBack, 173
In Interrupt::Idle, return true from CheckIfDue, 173
```

```
Switching from: add to: add
Now in thread: add
    interrupts: off -> on
```

(iv) machine/console.cc

```
void ConsoleOutput::PutChar(char ch)
```

- 模擬的IO裝置判斷putBusy，確保沒有其他的寫入操作正在執行
- 若沒有在執行中就可以進行寫入file
- 將putBusy設成TRUE
- 排程一個100 ticks (ConsoleTime) 後的ConsoleWriteInt，該interrupt執行完後會呼叫ConsoleOutput::CallBack()

(v) machine/interrupt.cc

```
void Interrupt::Schedule(CallBackObj *toCall, int fromNow, IntType type)
```

- 宣告when這個變數用以表示該 interrupt應該發生的時間點
- 宣告一個PendingInterrupt物件toOccur，這個物件代表即將發生的interrupt，記錄著發生interrupt時需要回呼的對象、時間及interrupt的類型
- 確fromNow大於0，因為若等於0就表示當下立刻就要執行中斷，但我們並不知道當前是否有中斷正在執行，以此避免衝突
- 將toOccur插入pending這個SortedList中並等待至該中斷應發生的時刻

(vi) machine/mipssim.cc

```
void Machine::Run()
```

- 在無窮迴圈中執行Machine::OneInstruction，執行指令後若未exit，則呼叫Interrupt::OneTick()

(vii) machine/interrupt.cc

```
void Interrupt::OneTick()
```

- 使kernel的時間前進SystemTick (10) 或UserTick (1)，取決於當下是在UserMode或SystemMode
- 為了要去檢查是否當前有需要執行的interrupts，先將interrupt關掉，避免處理interrupts時又有其他interrupts產生
- 進入CheckIfDue去檢查是否目前有需要執行的pending interrupt
- 檢查完後再透過ChangeLevel(IntOn, IntOff)重新打開interrupts
- 如果目前可以進行context switch (yieldOnReturn為TRUE)，則將yieldOnReturn設為FALSE，進入SystemMode後呼叫kernel->currentThread->Yield()，當此執行緒又可以執行時，則恢復Interrupt::status

(viii) machine/interrupt.cc

bool Interrupt::CheckIfDue(bool advanceClock)

- 確認目前處於關閉interrupt的狀態
- 若有設定interrupt除錯旗標，將呼叫DumpState()印出目前pending interrupts
- 透過next這個指標指向在pending interrupts中最接近目前時間的 interrupt，如果next的發生時間還沒達到，使用advanceClock判斷目前需不需要將totalTicks往後移到next指向的interrupt的發生時間，若不需要，則回傳FALSE，表示沒有任何的interrupt handler被啟動
- 若next的發生時間已經到達，且kernel->machine未被移除，則呼叫Machine::DelayedLoad模擬硬體的delayed load effect
- 將inHandler設成TRUE表示正在處理interrupt
- 取出pending interrupts中最前面的interrupt，並呼叫對應的interrupt handler (callback function) 進行處理，直到pending interrupts為空或是下一個interrupt發生時間大於目前的時間
- 做完後將inHandler設為FALSE表示處理完interrupts
- 回傳TRUE表示在呼叫本次CheckIfDue中至少處理一個interrupt

(ix) machine/console.cc

void ConsoleOutput::CallBack()

- 寫入動作結束並將putBusy設為FALSE
- 將console印出字元數統計量+1
- 呼叫SynchConsoleOutput::CallBack()

(x) userprog/synchconsole.cc

void SynchConsoleOutput::CallBack()

- 呼叫SynchConsoleOutput::waitFor的V()方法
- 禁用interrupt
- 若等待SynchConsoleOutput::waitFor的thread queue不為空，則取出最前面的thread，將其設為READY狀態，從以下add程式輸出的除錯訊息可以看到add執行緒被改回READY狀態，隨時準備再被CPU執行

```
In ConsoleOutput::CallBack(), 283
In SynchConsoleOutput::CallBack(), 283
In Semaphore::V(), 283
    interrupts: off -> off
Putting thread on ready list: add
    interrupts: off -> off
```

- 恢復interrupt level

(d) Makefile (code/test/Makefile)

```
include Makefile.dep

CC = $(GCCDIR)gcc
AS = $(GCCDIR)as
LD = $(GCCDIR)ld

INCDIR = -I../userprog -I../lib
CFLAGS = -g -G 0 -c $(INCDIR)
-B/usr/bin/local/nachos/lib/gcc-lib/decstation-ultrix/2.95.2/
-B/usr/bin/local/nachos/decstation-ultrix/bin/
```

- Makefile.dep包含所有依賴機器的定義
- CC、AS、LD分別指Compiler、Assembler及Linker
- INCDIR指出Header檔的目錄
- CFLAGS中-g表示保留除錯訊息，-c表示生成.o檔而不經由Linker連結成可執行檔，-B表示必須包含這些目錄下的binary檔或library¹

```
ifeq ($(hosttype), unknown)
PROGRAMS = unknownhost
else
# change this if you create a new test program!
#PROGRAMS = add halt shell matmult sort segments test1 test2 a
PROGRAMS = add halt createFile fileIO_test1 fileIO_test2
LotOfAdd
endif

all: $(PROGRAMS)
```

- 根據host機器類型定義PROGRAMS變量
- 定義Makefile預設目標all的依賴為PROGRAMS變量中的內容

```
start.o: start.S ../userprog/syscall.h
    $(CC) $(CFLAGS) $(ASFLAGS) -c start.S
```

- 定義start.o目標依賴於start.S組合語言檔及syscall.h標頭檔
- 定義Compiler如何將start.S組合語言檔編譯為start.o檔，其中-c表示生成.o檔而不經由Linker連結成可執行檔

```
halt.o: halt.c
    $(CC) $(CFLAGS) -c halt.c
halt: halt.o start.o
```

¹ <https://gcc.gnu.org/onlinedocs/gcc/Option-Index.html>

```
$(LD) $(LDFLAGS) start.o halt.o -o halt.coff
$(COFF2NOFF) halt.coff halt
```

- 定義halt.o由halt.c編譯而成
- 定義可執行檔halt依賴於halt.o及start.o
- 定義Linker將halt.o及start.o連結為halt.coff檔，又因為Nachos預設啟動程序（startup routine）位於0，所以第一個被傳入Linker的.o檔必須為start.o
- 使用COFF2NOFF將halt.coff轉為noff格式的Nachos可執行檔halt
- 其餘程式編譯步驟與halt相似，都是先編譯為.o檔，由Linker連結為.coff檔，再轉換成noff格式檔

```
clean:
    $(RM) -f *.o *.ii
    $(RM) -f *.coff
```

- make clean將會清除這些編譯生成的臨時檔

```
distclean: clean
    $(RM) -f $(PROGRAMS)
```

- make distclean將會清除這些Nachos可執行檔

```
unknownhost:
    @echo Host type could not be determined.
    @echo make is terminating.
    @echo If you are on an MFCF machine, contact the instructor
to report this problem
    @echo Otherwise, edit Makefile.dep and try again.
```

- 若host機器類型未知，印出終止訊息

Part II-2 I/O System Calls Implementation

我們在userprog/syscall.h定義SC_Open、SC_Read、SC_Write及SC_Close等新的system calls的列舉值，在test/start.S以組合語言定義對應的system call stubs，最後我們在userprog/exception.cc新增對應上述system calls的cases，以上system calls的操作十分相似，大致分為以下幾個步驟：

1. 讀入所需的arguments（存放在4至7號暫存器）
2. 呼叫對應的system call
3. 將system call回傳的結果寫回2號暫存器
4. 更新program counter

接下來將針對每個system call說明其中的實作細節

(a) OpenFileId Open(char *name)

- (i) SysOpen實作主要依靠FileSystem::OpenAFile方法，呼叫時傳入檔案名稱

```
OpenFileId OpenAFile(char *name)
{
    OpenFileId id;
    for (id = 0; id < 20; ++id)
    {
        if (OpenFileTable[id] == NULL)
        {
            break;
        }
    }
    // exceed the opened file limit
    if (id == 20)
    {
        return -1;
    }
    // call UNIX open and check
    int fileDescriptor = OpenForReadWrite(name, FALSE);
    if (fileDescriptor == -1)
    {
        return -1;
    }
    OpenFileTable[id] = new OpenFile(fileDescriptor);
    return id;
}
```

- (ii) 先檢查OpenFileTable是否有可用的entry，若無則回傳-1

- (iii) 透過sysdep.cc的OpenForReadWrite方法，在host的UNIX機器上以讀寫模式打開檔案並回傳file descriptor²
- (iv) 檢查file descriptor是否為-1，若是則開檔失敗，回傳-1
- (v) 若file descriptor不是-1，開檔成功，建立新的OpenFile entry並放入OpenFileTable中
- (vi) 回傳該OpenFile entry在OpenFileTable中的索引值

(b) int Write(char *buffer, int size, OpenFileId id)

- (i) SysWrite實作主要依靠FileSystem::WriteFile_方法，呼叫時傳入寫入來源buffer、寫入大小及欲寫入的OpenFileId

```
int WriteFile_(char *buffer, int size, OpenFileId id)
{
    if (buffer != NULL
        && size >= 0
        && is_valid_file_id(id))
    {
        return OpenFileTable[id]->Write(buffer, size);
    }
    return -1;
}

bool is_valid_file_id(OpenFileId id)
{
    return id >= 0 && id < 20 && OpenFileTable[id] != NULL;
}
```

- (ii) 檢查buffer、OpenFileId及寫入大小，OpenFileId需要在合法範圍內，且OpenFileTable確實已經開啟該檔，否則返回-1表示非法存取
- (iii) 針對該OpenFile entry呼叫OpenFile::Write方法，透過sysdep.cc的Lseek³及WriteFile⁴方法將檔案寫入host的UNIX機器，此操作後，OpenFile的currentOffset將被更新
- (iv) 成功返回寫入的大小

² <https://man7.org/linux/man-pages/man2/open.2.html>

³ <https://man7.org/linux/man-pages/man2/lseek.2.html>

⁴ <https://man7.org/linux/man-pages/man2/write.2.html>

(c) int Read(char *buffer, int size, OpenFileId id)

- (i) SysRead實作主要依靠FileSystem::ReadFile方法，呼叫時傳入讀取目的buffer、讀取大小及欲讀取的OpenFileId

```
int ReadFile(char *buffer, int size, OpenFileId id)
{
    if (buffer != NULL
        && size >= 0
        && is_valid_file_id(id))
    {
        return OpenFileTable[id]->Read(buffer, size);
    }
    return -1;
}
```

- (ii) 檢查buffer、OpenFileId及讀取大小，OpenFileId需要在合法範圍內，且OpenFileTable確實已經開啟該檔，否則返回-1表示非法存取
- (iii) 針對該OpenFile entry呼叫OpenFile::Read方法，透過sysdep.cc的Lseek及ReadPartial⁵方法從host的UNIX機器讀取檔案，並將讀取內容寫入目的buffer中，此操作後，OpenFile的currentOffset將被更新
- (iv) 成功返回讀取的大小

(d) int Close(OpenFileId id)

- (i) SysClose實作主要依靠FileSystem::CloseFile方法，呼叫時傳入欲關閉的OpenFileId

```
int CloseFile(OpenFileId id)
{
    if (is_valid_file_id(id))
    {
        delete OpenFileTable[id];
        OpenFileTable[id] = NULL;
        return 1;
    }
    return -1;
}
```

- (ii) 同樣先檢查OpenFileId是否在合法範圍內，且OpenFileTable確實已經開啟該檔，否則返回-1表示非法存取

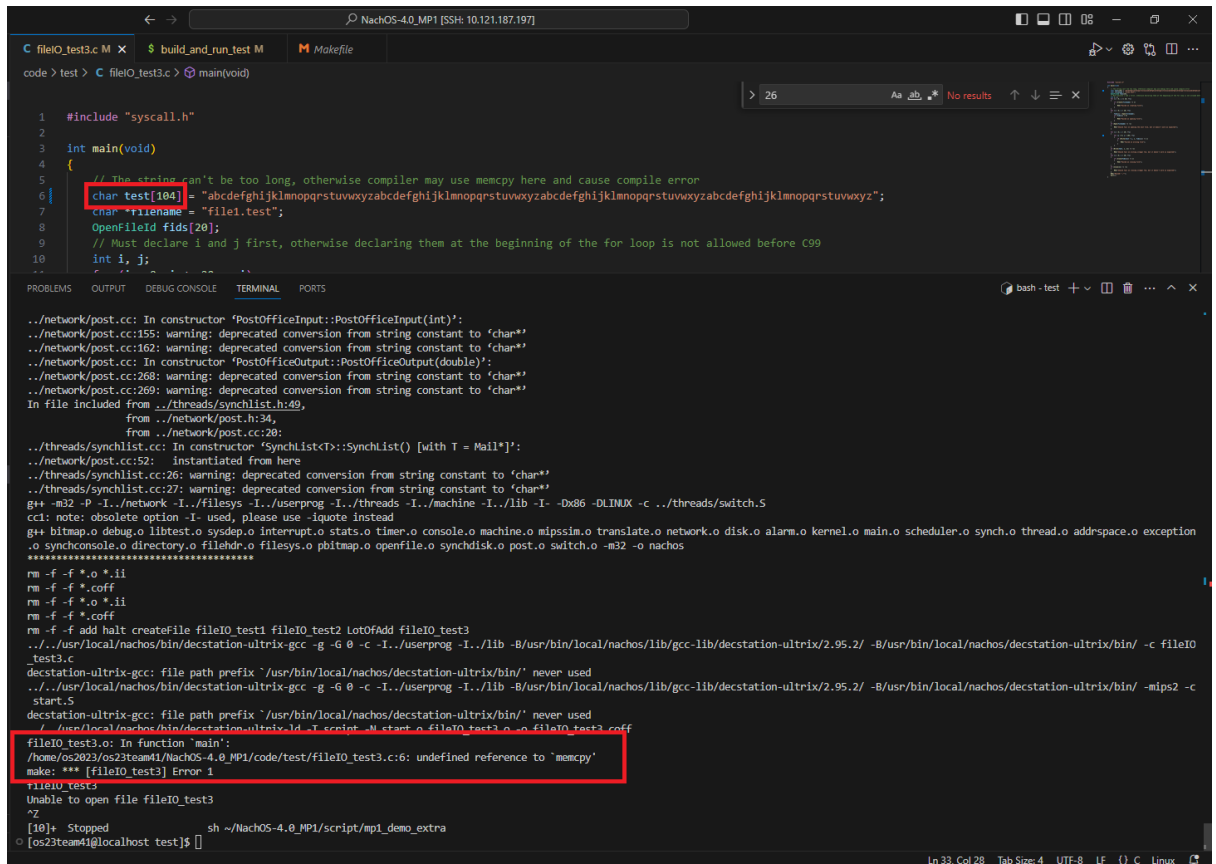
⁵ <https://man7.org/linux/man-pages/man2/read.2.html>

- (iii) 釋放該OpenFileId在OpenFileTable中所占用的entry，OpenFile的解構式呼叫sysdep.cc的Close⁶方法，在host的UNIX機器上關閉檔案
- (iv) 將該OpenFileId在OpenFileTable的位置設為NULL
- (v) 返回1表示成功

⁶ <https://man7.org/linux/man-pages/man2/close.2.html>

Difficulties encountered

- 在撰寫新的測試案例時，發現若是將IO test的字串長度改成原有案例的4倍（104個字元），編譯時會出現如下圖所示的undefined reference to 'memcpy'錯誤，似乎gcc在編譯時，有機會新增對memcpy的調用



```
code > test > C fileIO_test3.c > main(void)

1 #include "syscall.h"
2
3 int main(void)
4 {
5     // The string can't be too long, otherwise compiler may use memcpy here and cause compile error
6     char test[104] = "abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz";
7     char *filename = "file1.test";
8     OpenFileId fids[20];
9     // Must declare i and j first, otherwise declaring them at the beginning of the for loop is not allowed before C99
10    int i, j;
11
12    ...
13
14    ...
15
16    ...
17
18    ...
19
20    ...
21
22    ...
23
24    ...
25
26    ...
27
28    ...
29
30    ...
31
32    ...
33
34    ...
35
36    ...
37
38    ...
39
40    ...
41
42    ...
43
44    ...
45
46    ...
47
48    ...
49
50    ...
51
52    ...
53
54    ...
55
56    ...
57
58    ...
59
60    ...
61
62    ...
63
64    ...
65
66    ...
67
68    ...
69
70    ...
71
72    ...
73
74    ...
75
76    ...
77
78    ...
79
80    ...
81
82    ...
83
84    ...
85
86    ...
87
88    ...
89
90    ...
91
92    ...
93
94    ...
95
96    ...
97
98    ...
99
100   ...
101
102   ...
103
104   ...
105
106   ...
107
108   ...
109
110   ...
111
112   ...
113
114   ...
115
116   ...
117
118   ...
119
120   ...
121
122   ...
123
124   ...
125
126   ...
127
128   ...
129
130   ...
131
132   ...
133
134   ...
135
136   ...
137
138   ...
139
140   ...
141
142   ...
143
144   ...
145
146   ...
147
148   ...
149
150   ...
151
152   ...
153
154   ...
155
156   ...
157
158   ...
159
160   ...
161
162   ...
163
164   ...
165
166   ...
167
168   ...
169
170   ...
171
172   ...
173
174   ...
175
176   ...
177
178   ...
179
180   ...
181
182   ...
183
184   ...
185
186   ...
187
188   ...
189
190   ...
191
192   ...
193
194   ...
195
196   ...
197
198   ...
199
200   ...
201
202   ...
203
204   ...
205
206   ...
207
208   ...
209
210   ...
211
212   ...
213
214   ...
215
216   ...
217
218   ...
219
220   ...
221
222   ...
223
224   ...
225
226   ...
227
228   ...
229
230   ...
231
232   ...
233
234   ...
235
236   ...
237
238   ...
239
240   ...
241
242   ...
243
244   ...
245
246   ...
247
248   ...
249
250   ...
251
252   ...
253
254   ...
255
256   ...
257
258   ...
259
260   ...
261
262   ...
263
264   ...
265
266   ...
267
268   ...
269
270   ...
271
272   ...
273
274   ...
275
276   ...
277
278   ...
279
280   ...
281
282   ...
283
284   ...
285
286   ...
287
288   ...
289
290   ...
291
292   ...
293
294   ...
295
296   ...
297
298   ...
299
300   ...
301
302   ...
303
304   ...
305
306   ...
307
308   ...
309
310   ...
311
312   ...
313
314   ...
315
316   ...
317
318   ...
319
320   ...
321
322   ...
323
324   ...
325
326   ...
327
328   ...
329
330   ...
331
332   ...
333
334   ...
335
336   ...
337
338   ...
339
340   ...
341
342   ...
343
344   ...
345
346   ...
347
348   ...
349
350   ...
351
352   ...
353
354   ...
355
356   ...
357
358   ...
359
360   ...
361
362   ...
363
364   ...
365
366   ...
367
368   ...
369
370   ...
371
372   ...
373
374   ...
375
376   ...
377
378   ...
379
380   ...
381
382   ...
383
384   ...
385
386   ...
387
388   ...
389
390   ...
391
392   ...
393
394   ...
395
396   ...
397
398   ...
399
400   ...
401
402   ...
403
404   ...
405
406   ...
407
408   ...
409
410   ...
411
412   ...
413
414   ...
415
416   ...
417
418   ...
419
420   ...
421
422   ...
423
424   ...
425
426   ...
427
428   ...
429
430   ...
431
432   ...
433
434   ...
435
436   ...
437
438   ...
439
440   ...
441
442   ...
443
444   ...
445
446   ...
447
448   ...
449
450   ...
451
452   ...
453
454   ...
455
456   ...
457
458   ...
459
460   ...
461
462   ...
463
464   ...
465
466   ...
467
468   ...
469
470   ...
471
472   ...
473
474   ...
475
476   ...
477
478   ...
479
480   ...
481
482   ...
483
484   ...
485
486   ...
487
488   ...
489
490   ...
491
492   ...
493
494   ...
495
496   ...
497
498   ...
499
500   ...
501
502   ...
503
504   ...
505
506   ...
507
508   ...
509
510   ...
511
512   ...
513
514   ...
515
516   ...
517
518   ...
519
520   ...
521
522   ...
523
524   ...
525
526   ...
527
528   ...
529
530   ...
531
532   ...
533
534   ...
535
536   ...
537
538   ...
539
540   ...
541
542   ...
543
544   ...
545
546   ...
547
548   ...
549
550   ...
551
552   ...
553
554   ...
555
556   ...
557
558   ...
559
560   ...
561
562   ...
563
564   ...
565
566   ...
567
568   ...
569
570   ...
571
572   ...
573
574   ...
575
576   ...
577
578   ...
579
580   ...
581
582   ...
583
584   ...
585
586   ...
587
588   ...
589
590   ...
591
592   ...
593
594   ...
595
596   ...
597
598   ...
599
600   ...
601
602   ...
603
604   ...
605
606   ...
607
608   ...
609
610   ...
611
612   ...
613
614   ...
615
616   ...
617
618   ...
619
620   ...
621
622   ...
623
624   ...
625
626   ...
627
628   ...
629
630   ...
631
632   ...
633
634   ...
635
636   ...
637
638   ...
639
640   ...
641
642   ...
643
644   ...
645
646   ...
647
648   ...
649
650   ...
651
652   ...
653
654   ...
655
656   ...
657
658   ...
659
660   ...
661
662   ...
663
664   ...
665
666   ...
667
668   ...
669
670   ...
671
672   ...
673
674   ...
675
676   ...
677
678   ...
679
680   ...
681
682   ...
683
684   ...
685
686   ...
687
688   ...
689
690   ...
691
692   ...
693
694   ...
695
696   ...
697
698   ...
699
700   ...
701
702   ...
703
704   ...
705
706   ...
707
708   ...
709
710   ...
711
712   ...
713
714   ...
715
716   ...
717
718   ...
719
720   ...
721
722   ...
723
724   ...
725
726   ...
727
728   ...
729
730   ...
731
732   ...
733
734   ...
735
736   ...
737
738   ...
739
740   ...
741
742   ...
743
744   ...
745
746   ...
747
748   ...
749
750   ...
751
752   ...
753
754   ...
755
756   ...
757
758   ...
759
760   ...
761
762   ...
763
764   ...
765
766   ...
767
768   ...
769
770   ...
771
772   ...
773
774   ...
775
776   ...
777
778   ...
779
780   ...
781
782   ...
783
784   ...
785
786   ...
787
788   ...
789
790   ...
791
792   ...
793
794   ...
795
796   ...
797
798   ...
799
800   ...
801
802   ...
803
804   ...
805
806   ...
807
808   ...
809
810   ...
811
812   ...
813
814   ...
815
816   ...
817
818   ...
819
820   ...
821
822   ...
823
824   ...
825
826   ...
827
828   ...
829
830   ...
831
832   ...
833
834   ...
835
836   ...
837
838   ...
839
840   ...
841
842   ...
843
844   ...
845
846   ...
847
848   ...
849
850   ...
851
852   ...
853
854   ...
855
856   ...
857
858   ...
859
860   ...
861
862   ...
863
864   ...
865
866   ...
867
868   ...
869
870   ...
871
872   ...
873
874   ...
875
876   ...
877
878   ...
879
880   ...
881
882   ...
883
884   ...
885
886   ...
887
888   ...
889
890   ...
891
892   ...
893
894   ...
895
896   ...
897
898   ...
899
900   ...
901
902   ...
903
904   ...
905
906   ...
907
908   ...
909
910   ...
911
912   ...
913
914   ...
915
916   ...
917
918   ...
919
920   ...
921
922   ...
923
924   ...
925
926   ...
927
928   ...
929
930   ...
931
932   ...
933
934   ...
935
936   ...
937
938   ...
939
940   ...
941
942   ...
943
944   ...
945
946   ...
947
948   ...
949
950   ...
951
952   ...
953
954   ...
955
956   ...
957
958   ...
959
960   ...
961
962   ...
963
964   ...
965
966   ...
967
968   ...
969
970   ...
971
972   ...
973
974   ...
975
976   ...
977
978   ...
979
980   ...
981
982   ...
983
984   ...
985
986   ...
987
988   ...
989
990   ...
991
992   ...
993
994   ...
995
996   ...
997
998   ...
999
1000  ...
1001
1002  ...
1003
1004  ...
1005
1006  ...
1007
1008  ...
1009
1010  ...
1011
1012  ...
1013
1014  ...
1015
1016  ...
1017
1018  ...
1019
1020  ...
1021
1022  ...
1023
1024  ...
1025
1026  ...
1027
1028  ...
1029
1030  ...
1031
1032  ...
1033
1034  ...
1035
1036  ...
1037
1038  ...
1039
1040  ...
1041
1042  ...
1043
1044  ...
1045
1046  ...
1047
1048  ...
1049
1050  ...
1051
1052  ...
1053
1054  ...
1055
1056  ...
1057
1058  ...
1059
1060  ...
1061
1062  ...
1063
1064  ...
1065
1066  ...
1067
1068  ...
1069
1070  ...
1071
1072  ...
1073
1074  ...
1075
1076  ...
1077
1078  ...
1079
1080  ...
1081
1082  ...
1083
1084  ...
1085
1086  ...
1087
1088  ...
1089
1090  ...
1091
1092  ...
1093
1094  ...
1095
1096  ...
1097
1098  ...
1099
1100  ...
1101
1102  ...
1103
1104  ...
1105
1106  ...
1107
1108  ...
1109
1110  ...
1111
1112  ...
1113
1114  ...
1115
1116  ...
1117
1118  ...
1119
1120  ...
1121
1122  ...
1123
1124  ...
1125
1126  ...
1127
1128  ...
1129
1130  ...
1131
1132  ...
1133
1134  ...
1135
1136  ...
1137
1138  ...
1139
1140  ...
1141
1142  ...
1143
1144  ...
1145
1146  ...
1147
1148  ...
1149
1150  ...
1151
1152  ...
1153
1154  ...
1155
1156  ...
1157
1158  ...
1159
1160  ...
1161
1162  ...
1163
1164  ...
1165
1166  ...
1167
1168  ...
1169
1170  ...
1171
1172  ...
1173
1174  ...
1175
1176  ...
1177
1178  ...
1179
1180  ...
1181
1182  ...
1183
1184  ...
1185
1186  ...
1187
1188  ...
1189
1190  ...
1191
1192  ...
1193
1194  ...
1195
1196  ...
1197
1198  ...
1199
1200  ...
1201
1202  ...
1203
1204  ...
1205
1206  ...
1207
1208  ...
1209
1210  ...
1211
1212  ...
1213
1214  ...
1215
1216  ...
1217
1218  ...
1219
1220  ...
1221
1222  ...
1223
1224  ...
1225
1226  ...
1227
1228  ...
1229
1230  ...
1231
1232  ...
1233
1234  ...
1235
1236  ...
1237
1238  ...
1239
1240  ...
1241
1242  ...
1243
1244  ...
1245
1246  ...
1247
1248  ...
1249
1250  ...
1251
1252  ...
1253
1254  ...
1255
1256  ...
1257
1258  ...
1259
1260  ...
1261
1262  ...
1263
1264  ...
1265
1266  ...
1267
1268  ...
1269
1270  ...
1271
1272  ...
1273
1274  ...
1275
1276  ...
1277
1278  ...
1279
1280  ...
1281
1282  ...
1283
1284  ...
1285
1286  ...
1287
1288  ...
1289
1290  ...
1291
1292  ...
1293
1294  ...
1295
1296  ...
1297
1298  ...
1299
1300  ...
1301
1302  ...
1303
1304  ...
1305
1306  ...
1307
1308  ...
1309
1310  ...
1311
1312  ...
1313
1314  ...
1315
1316  ...
1317
1318  ...
1319
1320  ...
1321
1322  ...
1323
1324  ...
1325
1326  ...
1327
1328  ...
1329
1330  ...
1331
1332  ...
1333
1334  ...
1335
1336  ...
1337
1338  ...
1339
1340  ...
1341
1342  ...
1343
1344  ...
1345
1346  ...
1347
1348  ...
1349
1350  ...
1351
1352  ...
1353
1354  ...
1355
1356  ...
1357
1358  ...
1359
1360  ...
1361
1362  ...
1363
1364  ...
1365
1366  ...
1367
1368  ...
1369
1370  ...
1371
1372  ...
1373
1374  ...
1375
1376  ...
1377
1378  ...
1379
1380  ...
1381
1382  ...
1383
1384  ...
1385
1386  ...
1387
1388  ...
1389
1390  ...
1391
1392  ...
1393
1394  ...
1395
1396  ...
1397
1398  ...
1399
1400  ...
1401
1402  ...
1403
1404  ...
1405
1406  ...
1407
1408  ...
1409
1410  ...
1411
1412  ...
1413
1414  ...
1415
1416  ...
1417
1418  ...
1419
1420  ...
1421
1422  ...
1423
1424  ...
1425
1426  ...
1427
1428  ...
1429
1430  ...
1431
1432  ...
1433
1434  ...
1435
1436  ...
1437
1438  ...
1439
1440  ...
1441
1442  ...
1443
1444  ...
1445
1446  ...
1447
1448  ...
1449
1450  ...
1451
1452  ...
1453
1454  ...
1455
1456  ...
1457
1458  ...
1459
1460  ...
1461
1462  ...
1463
1464  ...
1465
1466  ...
1467
1468  ...
1469
1470  ...
1471
1472  ...
1473
1474  ...
1475
1476  ...
1477
1478  ...
1479
1480  ...
1481
1482  ...
1483
1484  ...
1485
1486  ...
1487
1488  ...
1489
1490  ...
1491
1492  ...
1493
1494  ...
1495
1496  ...
1497
1498  ...
1499
1500  ...
1501
1502  ...
1503
1504  ...
1505
1506  ...
1507
1508  ...
1509
1510  ...
1511
1512  ...
1513
1514  ...
1515
1516  ...
1517
1518  ...
1519
1520  ...
1521
1522  ...
1523
1524  ...
1525
1526  ...
1527
1528  ...
1529
1530  ...
1531
1532  ...
1533
1534  ...
1535
1536  ...
1537
1538  ...
1539
1540  ...
1541
1542  ...
1543
1544  ...
1545
1546  ...
1547
1548  ...
1549
1550  ...
1551
1552  ...
1553
1554  ...
1555
1556  ...
1557
1558  ...
1559
1560  ...
1561
1562  ...
1563
1564  ...
1565
1566  ...
1567
1568  ...
1569
1570  ...
1571
1572  ...
1573
1574  ...
1575
1576  ...
1577
1578  ...
1579
1580  ...
1581
1582  ...
1583
1584  ...
1585
1586  ...
1587
1588  ...
1589
1590  ...
1591
1592  ...
1593
1594  ...
1595
1596  ...
1597
1598  ...
1599
1600  ...
1601
1602  ...
1603
1604  ...
1605
1606  ...
1607
1608  ...
1609
1610  ...
1611
1612  ...
1613
1614  ...
1615
1616  ...
1617
1618  ...
1619
1620  ...
1621
1622  ...
1623
1624  ...
1625
1626  ...
1627
1628  ...
1629
1630  ...
1631
1632  ...
1633
1634  ...
1635
1636  ...
1637
1638  ...
1639
1640  ...
1641
1642  ...
1643
1644  ...
1645
1646  ...
1647
1648  ...
1649
1650  ...
1651
1652  ...
1653
1654  ...
1655
1656  ...
1657
1658  ...
1659
1660  ...
1661
1662  ...
1663
1664  ...
1665
1666  ...
1667
1668  ...
1669
1670  ...
1671
1672  ...
1673
1674  ...
1675
1676  ...
1677
1678  ...
1679
1680  ...
1681
1682  ...
1683
1684  ...
1685
1686  ...
1687
1688  ...
1689
1690  ...
1691
1692  ...
1693
1694  ...
1695
1696  ...
1697
1698  ...
1699
1700  ...
1701
1702  ...
1703
1704  ...
1705
1706  ...
1707
1708  ...
1709
1710  ...
1711
1712  ...
1713
1714  ...
1715
1716  ...
1717
1718  ...
1719
1720  ...
1721
1722  ...
1723
1724  ...
1725
1726  ...
1727
1728  ...
1729
1730  ...
1731
1732  ...
1733
1734  ...
1735
1736  ...
1737
1738  ...
1739
1740  ...
1741
1742  ...
1743
1744  ...
1745
1746  ...
1747
1748  ...
1749
1750  ...
1751
1752  ...
1753
1754  ...
1755
1756  ...
1757
1758  ...
1759
1760  ...
1761
1762  ...
1763
1764  ...
1765
1766  ...
1767
1768  ...
1769
1770  ...
1771
1772  ...
1773
1774  ...
1775
1776  ...
1777
1778  ...
1779
1780  ...
1781
1782  ...
1783
1784  ...
1785
1786  ...
1787
1788  ...
1789
1790  ...
1791
1792  ...
1793
1794  ...
1795
1796  ...
1797
1798  ...
1799
1800  ...
1801
1802  ...
1803
1804  ...
1805
1806  ...
1807
1808  ...
1809
1810  ...
1811
1812  ...
1813
1814  ...
1815
1816  ...
1817
1818  ...
1819
1820  ...
1821
1822  ...
1823
1824  ...
1825
1826  ...
1827
1828  ...
1829
1830  ...
1831
1832  ...
1833
1834  ...
1835
1836  ...
1837
1838  ...
1839
1840  ...
1841
1842  ...
1843
1844  ...
1845
1846  ...
1847
1848  ...
1849
1850  ...
1851
1852  ...
1853
1854  ...
1855
1856  ...
1857
1858  ...
1859
1860  ...
1861
1862  ...
1863
1864  ...
1865
1866  ...
1867
1868  ...
1869
1870  ...
1871
1872  ...
1873
1874  ...
1875
1876  ...
1877
1878  ...
1879
1880  ...
1881
1882  ...
1883
1884  ...
1885
1886  ...
1887
1888  ...
1889
1890  ...
1891
1892  ...
1893
1894  ...
1895
1896  ...
1897
1898  ...
1899
1900  ...
1901
1902  ...
1903
1904  ...
1905
1906  ...
1907
1908  ...
1909
1910  ...
1911
1912  ...
1913
1914  ...
1915
1916  ...
1917
1918  ...
1919
1920  ...
1921
1922  ...
1923
1924  ...
1925
1926  ...
1927
1928  ...
1929
1930  ...
1931
1932  ...
1933
1934  ...
1935
1936  ...
1937
1938  ...
1939
1940  ...
1941
1942  ...
1943
1944  ...
1945
1946  ...
1947
1948  ...
1949
1950  ...
1951
1952  ...
1953
1954  ...
1955
1956  ...
1957
1958  ...
1959
1960  ...
1961
1962  ...
1963
1964  ...
1965
1966  ...
1967
1968  ...
1969
1970  ...
1971
1972  ...
1973
1974  ...
1975
1976  ...
1977
1978  ...
1979
1980  ...
1981
1982  ...
1983
1984  ...
1985
1986  ...
1987
1988  ...
1989
1990  ...
1991
1992  ...
1993
1994  ...
1995
1996  ...
1997
1998  ...
1999
2000  ...
2001
2002  ...
2003
2004  ...
2005
2006  ...
2007
2008  ...
2009
2010  ...
2011
2012  ...
2013
2014  ...
2015
2016  ...
2017
2018  ...
2019
2020  ...
2021
2022  ...
2023
2024  ...
2025
2026  ...
2027
2028  ...
2029
2030  ...
2031
2032  ...
2033
2034  ...
2035
2036  ...
2037
2038  ...
2039
2040  ...
2041
2042  ...
2043
2044  ...
2045
2046  ...
2047
2048  ...
2049
2050  ...
2051
2052  ...
2053
2054  ...
2055
2056  ...
2057
2058  ...
2059
2060  ...
2061
2062  ...
2063
2064  ...
2065
2066  ...
2067
2068  ...
2069
2070  ...
2071
2072  ...
2073
2074  ...
2075
2076  ...
2077
2078  ...
2079
2080  ...
2081
2082  ...
2083
2084  ...
2085
2086  ...
2087
2088  ...
2089
2090  ...
2091
2092  ...
2093
2094  ...
2095
2096  ...
2097
2098  ...
2099
2100  ...
2101
2102  ...
2103
2104  ...
2105
2106  ...
2107
2108  ...
2109
2110  ...
2111
2112  ...
2113
2114  ...
2115
2116  ...
2117
2118  ...
2119
2120  ...
2121
2122  ...
2123
2124  ...
2125
2126  ...
2127
2128  ...
2129
2130  ...
2131
2132  ...
2133
2134  ...
2135
2136  ...
2137
2138  ...
2139
2140  ...
2141
2142  ...
2143
2144  ...
2145
2146  ...
2147
2148  ...
2149
2150  ...
2151
2152  ...
2153
2154  ...
2155
2156  ...
2157
2158  ...
2159
2160  ...
2161
2162  ...
2163
2164  ...
2165
2166  ...
2167
2168  ...
2169
2170  ...
2171
2172  ...
2173
2174  ...
2175
2176  ...
2177
2178  ...
2179
2180  ...
2181
2182  ...
2183
2184  ...
2185
2186  ...
2187
218
```

Feedback

- 建議compile時可以指定-std選項，或是提醒同學撰寫測試程式時不要用過新的C語言語法，我在撰寫測試程式時，花了一些時間除錯才發現for loop開頭宣告變數是C99之後才允許的⁷

⁷ <https://stackoverflow.com/questions/1287863/c-for-loop-int-initial-declaration>