

Lập trình FORTRAN

Nguyễn Hồng Quang

Khoa KT-TV-HDH

SĐT: 096 451 77 00

Email: quangnh@hus.edu.vn

Giới thiệu môn học

Mục tiêu khóa học

Sinh viên sau khi học xong khóa học này cần phải biết:

- Viết được một chương trình bằng ngôn ngữ lập trình FORTRAN.
- Dịch và chạy chương trình FORTRAN trên bằng trình dịch gfortran trên linux bằng dòng lệnh.

Phương pháp đánh giá

Quan niệm: Học là quá trình lao động tích lũy lâu dài, vì vậy nếu bạn đã chăm chỉ học trước đây, và đã đạt được yêu cầu môn học này đặt ra, vậy bây giờ bạn có thể nghỉ ngơi hay tiếp tục chăm chỉ học tập những thứ mà bạn chưa biết. Vì vậy:

- Việc tham gia nghe giảng trên lớp là không bắt buộc.
- Sinh viên vì lý do gì đó có thể rời bỏ lớp học bất cứ lúc nào.
- Yêu cầu sinh viên không làm ảnh hưởng tới việc giảng dạy của giáo viên và việc học tập của các sinh viên khác.

Phương pháp đánh giá

Quá trình kiểm tra, đánh giá được thực hiện online trên <https://physics.hus.edu.vn/moodle/>

Sinh viên phải đăng ký email với giáo viên để lấy tài khoản trên trang này.

- Hàng tuần, sinh viên phải làm kiểm tra thường xuyên tại nhà - có giới hạn thời gian làm bài.
- Sinh viên phải tự mang máy tính, và làm bài kiểm tra giữa kỳ, cuối kỳ trên trang này tại trường. Qua trình làm bài có sự giám sát của giáo viên.
- Điểm thi được tự động chấm dựa trên kết quả làm bài của sinh viên.

Đề nghị cho sinh viên

Để đảm bảo quá trình thực tập, đề nghị sinh viên cài đặt một trong các thứ sau:

Nếu có điều kiện - sinh viên có máy tính với ổ cứng đủ lớn, sinh viên nên dành riêng một phân vùng ổ cứng trên máy tính để cài một hệ điều hành linux nào đó - tốt nhất nên là Debian hay Ubuntu song song với hệ điều hành Windows. Để làm việc này, chọn một ổ cứng (D hay E) rồi sao chép toàn bộ dữ liệu sang các ổ cứng khác. Đây sẽ là phân vùng để cài linux. Tải một bộ cài linux - file .iso trên mạng về, ghi vào USB rồi khởi động từ USB này để cài đặt. Chuyển đổi giữa hai hệ điều hành được thực hiện bởi việc khởi động lại máy tính, lúc này một menu (thông thường là grub) sẽ cho phép chúng ta chọn HĐH được khởi động.

Đề nghị cho sinh viên

Trong trường hợp không thể dành ra một phân vùng riêng, nhưng còn dư một lượng ổ cứng đủ lớn (>20G), thì có thể sử dụng Hyper-V để cài đặt một máy ảo linux chạy trên nền HĐH Windows. Việc này có thể làm được trên các máy có HĐH Windows 10 Pro/Enterprise/Education, các HĐH này hỗ trợ sẵn Hyper-V, không cần cài thêm. Sau khi enable Hyper-V, khởi động lại máy, tải một đĩa cài linux dưới dạng .iso về, trên menu start, chọn Hyper-V Quick Create, theo các bước hướng dẫn để tạo và cài máy ảo linux. Nhớ là khi tạo máy ảo, chọn tệp .iso cài đặt linux vừa tải về như là ổ CD giả lập và khởi động máy ảo từ CD.

Đề nghị cho sinh viên

Trong trường hợp không thể làm được một trong hai việc trên thì cài đặt cygwin, một giả lập GNU trên windows. Tải từ trang <https://www.cygwin.com/> hay link https://www.cygwin.com/setup-x86_64.exe tiếp https://www.cygwin.com/setup-x86_64.exe. Chạy và chọn next cho đến khi kết thúc.

Đề nghị cho sinh viên

Cài trình dịch gfortran, gcc trên HĐH linux hay trên cygwin.

Để được hỗ trợ đề nghị liên hệ với thầy Nguyễn Hồng Quang - ĐT: 0964517700.

Chương trình là gì?

Chương trình là gì?

Trước khi học lập trình bằng ngôn ngữ FORTRAN, Chúng ta cần phải hiểu thế nào là chương trình. Và các kỹ năng gì cần phải có để lập trình.

Có thể hiểu một cách đơn giản, chương trình là một loạt các chỉ dẫn để máy tính thực hiện một công việc nào đó. Vậy cái gì là quan trọng nhất khi lập trình. Hãy tưởng tượng một lớp học ở Mỹ, ở đó giáo viên đang dạy các học viên cách nối hai sợi dây, và một lớp học khác tại đây, giáo viên cũng dạy các học viên nối hai sợi dây. Vậy hai học viên khi nối hai sợi dây (thực hiện chương trình) sẽ làm khác nhau hay cùng làm giống nhau? Tất nhiên câu trả lời là giống nhau, mặc dù giáo viên ở Mỹ truyền đạt các chỉ dẫn bằng tiếng Anh, trong khi ở đây, chúng ta dùng tiếng Việt.

Chương trình là gì?

Như vậy ngôn ngữ không quá quan trọng ở đây, mà quan trọng là các chỉ dẫn, mặc dù nó có thể được biểu diễn bằng ngôn ngữ tiếng Anh, ngôn ngữ tiếng Việt, hay ngôn ngữ FORTRAN. Các chỉ dẫn này được gọi là thuật toán. Như vậy, nếu như chúng ta có một thuật toán được biểu diễn bởi ngôn ngữ tiếng Việt và có một bộ dịch (kiểu như google dịch) có thể dịch từ ngôn ngữ tiếng Việt sang ngôn ngữ FORTRAN thì việc viết một chương trình bằng ngôn ngữ FORTRAN trở nên rất đơn giản. Tất nhiên khi ấy, chúng ta phải viết ra được thuật toán bằng ngôn ngữ tiếng Việt. Điều này là hiển nhiên, nếu bản thân giáo viên không biết cách nối hai sợi dây với nhau thì làm sao có thể chỉ dẫn cho các học viên cách nối chúng lại với nhau được?

Các kỹ năng cần thiết để lập trình

Như vậy các kỹ năng cần thiết để có thể lập trình FORTRAN được:

- 1)Biết cách giải quyết vấn đề mà đề bài yêu cầu.
Nói cách khác chúng ta có thể làm thủ công để giải quyết các yêu cầu đặt ra.
- 2)Biết cách biểu diễn các chỉ dẫn để giải quyết theo từng bước bằng ngôn ngữ tiếng Việt - viết ra được thuật toán giải ở trên dưới dạng các bước tuần tự.
- 3)Biết cách dịch các chỉ dẫn từ ngôn ngữ tiếng Việt sang ngôn ngữ FORTRAN - chính là môn học này.

Giới thiệu FORTRAN

Giới thiệu FORTRAN

Fortran (hay FORTRAN - FORMula TRANslating System) là một ngôn ngữ lập trình biên dịch, được phát triển từ thập niên 1950 bởi IBM và vẫn được dùng nhiều trong tính toán khoa học cho đến nay. Các phiên bản đầu có tên chính thức là FORTRAN, nhưng chữ hoa được chuyển sang chữ thường từ phiên bản Fortran 90. Tiêu chuẩn quốc tế cho tên gọi này ngày nay là "Fortran". Hiện nay, fortran hỗ trợ cả lập trình hướng đối tượng (fortran 2003) và lập trình song song (fortran 2008). Ngôn ngữ này được các nhà khoa học ưa chuộng do số thực trong fortran mặc nhiên là số phức - trong khi các ngôn ngữ khác thì đây là hai kiểu khác biệt.

Giới thiệu FORTRAN

FORTRAN đã được cải tiến qua nhiều thế hệ (FORTRAN I/II/III/IV, FORTRAN 66, FORTRAN 77, fortran 90,...). Nhưng phiên bản cũ nhất còn được sử dụng ngày nay là FORTRAN 77.

Chương trình FORTRAN 77 phải viết với khuôn dạng phức tạp, 6 ký tự đầu của mọi dòng được dành riêng, câu lệnh bắt đầu từ ký tự thứ 7. Dòng bắt đầu bởi ký tự C, c hay * được coi là dòng chú thích. 5 ký tự đầu nếu không phải là ký tự trống hay chú thích thì được coi là nhãn của câu lệnh. Ký tự thứ 6 là dấu + thể hiện dòng này là dòng tiếp tục của dòng lệnh trên. Việc này làm cho việc soạn thảo chương trình trở nên phức tạp hơn và chương trình cũng khó đọc hơn.

Giới thiệu FORTRAN

Fortran 90 là phiên bản đầu tiên bỏ các tiêu chuẩn này, các chương trình của fortran 90 được soạn thảo hoàn toàn tự do - câu lệnh có thể bắt đầu từ bất kỳ vị trí nào trên dòng. Tuy nhiên căn lề các câu lệnh một cách hợp lý sẽ làm cho chương trình trở nên dễ theo dõi hơn, giúp cho việc sửa lỗi chương trình trở nên đơn giản hơn rất nhiều. Nên nhớ rằng, **chương trình không phải chỉ để cho máy hiểu, chương trình là để cho người đọc.**

Giới thiệu FORTRAN

Fortran có nhiều trình dịch khác nhau, mỗi trình dịch có những sự khác biệt nho nhỏ. Trong môn học này chúng ta sử dụng trình dịch là gfortran (GNU fortran). Mặc dù có sự khác biệt với các trình dịch khác, nhưng với yêu cầu đơn giản của môn học này thì những sự khác biệt này sẽ không ảnh hưởng đến chúng ta. Ở đây, chúng ta chỉ cần chú ý rằng, một tên tệp chương trình nếu có đuôi là .f, .f77 .for thì gfortran sẽ tự động hiểu đây là chương trình là FORTRAN 77, đuôi .F, .FOR thì được chuyển qua một bộ tiền xử lý trước khi dịch, vì vậy nó có thể dùng khuôn dạng tự do. Đuôi .f90, f95 .f03 hay f08 dùng cho các phiên bản fortran 90, 95, 2003 và 2008. Fortran 90 có một số vấn đề, vì vậy tốt nhất chúng ta nên dùng đuôi là .f95.

Lập trình FORTRAN

chương trình FORTRAN đơn giản

Bắt đầu với một chương trình fortran đơn giản:

nano chao_cac_ban.f95

```
program chao_cac_ban  
  print *, 'chao cac ban'  
end program
```

gfortran chao_cac_ban.f95 -o chao_cac_ban
./chao_cac_ban

Cấu trúc chương trình Fortran

```
PROGRAM Tên_chương_trình
  Các lệnh thực hiện chương trình
END PROGRAM
```

Cấu trúc một chương trình cơ bản.

```
FUNCTION Tên_hàm(Danh, sách, biến)
  Các lệnh thực hiện hàm
  Tên_hàm = giá_trị_trả_về
RETURN
END FUNCTION Tên_hàm
```

Cấu trúc một hàm cơ bản.
Phần chữ xám là phần không nhất thiết phải có.

```
SUBROUTINE Tên_thủ_tục (Danh, sách, biến)
  Các lệnh thực hiện thủ tục
RETURN
END SUBROUTINE Tên_thủ_tục
```

Cấu trúc một thủ tục cơ bản.

Phân loại dụng chương trình Fortran

Khai báo các biến

Khai báo biến nên gom chung vào phần đầu chương trình để người ddochj dễ theo dõi.

Các lệnh khác:

Đọc dữ liệu đầu vào
Lệnh biến đổi các biến.
In các kết quả ra
.....

Các lệnh thực hiện chương trình.

Lệnh Fortran cơ bản - khai báo

Parameter (PI = 3.14159)	Khai báo hằng số.
real :: so_nguyen, mang_1(5) real*4 so_nguyen_1 real*4, dimention(5) :: mang_2	khai báo biến (mảng 5 phần tử) thực độ chính xác đơn. Số dấu chấm động 32 bit (10^{38} , 7 chữ số có nghĩa)
real*8 :: so_thuc, mang_1(5) double precision, dimention(5) :: mang_2	khai báo biến (mảng 5 phần tử) thực độ chính xác kép, số dấu chấm động 64 bit (10^{308} , 15 chữ số).
complex phuc_1 complex*8 :: phuc_2	khai báo biến phức độ chính xác đơn, kép.

Lệnh Fortran cơ bản - khai báo

<pre>character :: chu_a character (len=20) :: dong_1</pre>	khai báo biến ký tự - xâu ký tự 20 phần tử .
<pre>logical :: a logical, dimention(5) :: a</pre>	khai báo biến (mảng 5 phần tử) logic.
<pre>integer :: i, j(5) integer*4 k, l(5) dimension(5) :: m, n</pre>	khai báo biến (mảng 5 phần tử) nguyên. Khoảng giá trị 2.147.483.648.
<pre>integer*2 :: i, j(2), k(5) Integer*2, dimention(5) :: k</pre>	khai báo biến (mảng 5 phần tử) nguyên. Khoảng 32768.
<pre>Data i k(1) j /5 1 2*0/ Data (k(i), i=1,3) /3*0/</pre>	Khởi tạo i=5, k(1)=1, j=(0,0) Khởi tạo k(1 đến 3) = 0.

Fortran cơ bản - Các giá trị cơ bản

.True. .False.	Giá trị đúng – sai cho các biến dạng logic.
1 2	Giá trị số thể hiện các giá trị nguyên - dùng cho biến nguyên.
4. .4 1e2 3e-2	Giá trị số thể hiện các biến thực - dùng cho biến thực. Thể hiện các giá trị 4.0, 0.4, 100 ($1 * 10^2$), 0.03 ($3*10^{-2}$).
'a' "Thong bao" 'Thong bao'	Giá trị ký tự, xâu ký tự.

Lệnh Fortran cơ bản - Điều khiển

```
if (điều kiện) then  
    Lệnh  
else  
    Lệnh  
end if
```

Lệnh rẽ nhánh theo điều kiện

```
do nhãn  
i=đầu,cuối,bước  
    Lệnh  
nhãn continue  
end do
```

Lệnh lặp xác định cho f77 (f90)
với fortran 77, dùng màu đỏ, không dùng màu xanh.
fortran 90 ngược lại.

```
do while (điều kiện)  
    Lệnh  
    if (điều kiện) exit  
end do
```

Lệnh lặp không xác định (f90).
Dùng màu đỏ xác định điều kiện trước. màu xanh xác định điều kiện sau.

Lệnh Fortran cơ bản - Đọc ghi

Read (*,*) a, b Read (ID,format) a	Lệnh đọc các biến. đọc từ file/thiết bị (unit) theo khuôn dạng.
Write(*,*,option) a,b Write(ID,format) a	lệnh in các biến theo khuôn dạng ghi ra file/thiết bị theo khuôn dạng.
Print * Print format	lệnh in các biến theo khuôn dạng ra màn hình.
Nhãn format “khuôn dạng”	Lệnh khai báo khuôn dạng đọc/ghi ra màn hình cho lệnh Read/Write

Lệnh Fortran cơ bản - Khuôn dạng

I w.m	Số nguyên tối thiểu m số, tối đa w số.
F w.d E [S N]w.d E e	Số thực, tối đa w số, d số sau dấu phẩy, e số cho phần mũ.
L w	biến logical
A w	Ký tự / chuỗi ký tự
Label	format ([“thông báo”,]khuôn dạng[,...])

Chữ màu đỏ thể hiện kiểu số liệu cần in.	w: Tổng số ký tự sẽ in ra / đọc vào.
Chữ màu đen thể hiện tham số bắt buộc phải có.	d: Số ký tự cho phần thập phân.
Chữ màu xám thể hiện tham số không bắt buộc.	e: Số ký tự cho phần mũ.

Lệnh Fortran cơ bản - Khuôn dạng

\	Không xuống dòng
/	Xuống dòng
wX	In w dấu trống.
TLw	Nhảy sang trái w ký tự (từ vị trí hiện tại).
TRw	Nhảy sang phải w ký tự.
Label	format ([“thông báo”,]khuôn dạng[,...])

Chữ màu đen thể hiện tham số bắt buộc phải có.

Chữ màu xám thể hiện tham số không bắt buộc.

Toán tử Fortran cơ bản - Logic

a .and. b	a và b
a .or. b	a hoặc b
.not. b	negation đảo b
a .eqv. b	a giống b
a .neqv. b	a không giống b

Toán tử Fortran cơ bản - so sánh

a == b	a .eq. b	a bằng b ?
a /= b	a .ne. b	a không bằng b ?
a > b	a .gt. b	a lớn hơn b ?
a < b	a .lt. b	a nhỏ hơn b?
a >= b	a .ge. b	a lớn hơn hoặc bằng b?
a <= b	a .le. b	a nhỏ hơn hoặc bằng b?

Toán tử Fortran cơ bản - số học

$a + b$

a cộng b

$a - b$

a trừ b

$a * b$

a nhân b

a / b

a chia b

$a ** b$

a mũ b

Lệnh với tệp

**OPEN (ID, FILE='Ten_tệp',
STATUS='old', ERR=nhãn,
IOSTAT=biến_nguyên,
ACTION=kiểu_tệp)**

mở tệp

CLOSE(ID)

đóng tệp

STATUS có thể là: **old** (tệp đã tồn tại), **new** (tạo mới), **replace** (xóa rồi tạo mới nếu đã tồn tại), **unknown** (mặc định - nếu chưa tồn tại thì mở mới), **scratch** (tạo mới và xóa đi khi đóng lại - dùng để ghi tạm). **IOSTAT** là trạng thái trả về khi thực hiện lệnh với tệp (có trong mọi lệnh với tệp) là số nguyên, **bằng 0** khi không có lỗi, **khác 0** là có lỗi. **ERR** khai báo nhãn lệnh nhảy đến khi có lỗi. **ACTION** có thể là **read** (tệp chỉ đọc - dùng với tệp đã tồn tại), **write** (tệp chỉ ghi), **readwrite** (dùng để đọc ghi). Có thể thay thế **ID** bằng tham số **NEWUNIT=biến_ID**, trng đó **biến_ID** là **biến nguyên**, để đảm bảo không bị trùng ID với các thiết bị đã tồn tại, giá trị trong biến_ID được hệ thống tự động gán khi chạy chương trình.

Lệnh với tệp

**READ(ID, *, IOSTAT=biếntrạngthái)
biến**

đọc tệp

READ(ID, *, END=nhãn) biến

READ(ID, *, ERR=nhãn) biến

READ(ID, format, END=nhãn) biến

READ(ID, *) (biến_mảng(i,j) i=1,10)

Với lệnh đọc đầu, trạng thái lỗi (thường xuất hiện ở cuối tệp) được trả về trong biến trạng thái, dùng biến này để kết thúc quá trình đọc. Lệnh sau nhảy đến nhãn nếu đến cuối tệp. Lệnh cuối đọc 10 giá trị (trên 1 dòng) vào biến biến_mảng theo cột j, các giá trị hàng i từ 1 đến 10.

Lệnh với chuỗi ký tự

Xâu_2 = Xâu_1(3:5) Xâu_2 = Xâu_1(:5) Xâu_2 = Xâu_1(3:)	trích xâu con, xâu con chỉ có thể nằm bên phải dấu gán.
Xâu_3 = Xâu_2 // “,” // Xâu_1(1:7)	nối các xâu
READ(Xâu, format) biến WRITE(Xâu, format) biến	Đọc / ghi từ / đến xâu ký tự thay vì thiết bị.

Hàm ký tự chuẩn

LEN(Xâu)	Trả về độ dài xâu.
INDEX(Xâu_con, Xâu_chính)	Trả về vị trí của xâu con trong xâu chính.
ACHAR(ký_tự)	Trả về vị trí của ký_tự trong bảng mã ASCII.
IACHAR(Giá_trị)	Trả về ký tự thứ giá_trị trong bảng mã ASCII.
SCAN(xâu, tập_ký_tự, từ_trái)	Trả về vị trí đầu tiên xuất hiện bất kỳ ký tự nào của tập_ký_tự trong xâu.
VERIFY(xâu, tập_ký_tự)	Trả về vị trí đầu tiên không xuất hiện ký tự của tập_ký_tự trong xâu.

Hàm ký tự chuẩn

TRIM(Xâu)	Trả về xâu bị loại bỏ các dấu trống ở đầu Xâu.
LEN_TRIM(Xâu)	Tương đương LEN(TRIM(Xâu))
REPEAT(Xâu, n)	Trả về xâu gồm n lần lặp xâu.
ADJUSTL(Xâu) ADJUSTR(Xâu)	Trả về xâu được căn trái / phải. Các dấu trống được dồn về phải / trái.

Thank you!