فصل پنجم
ورودی و خروجی

سید ناصر رضوی

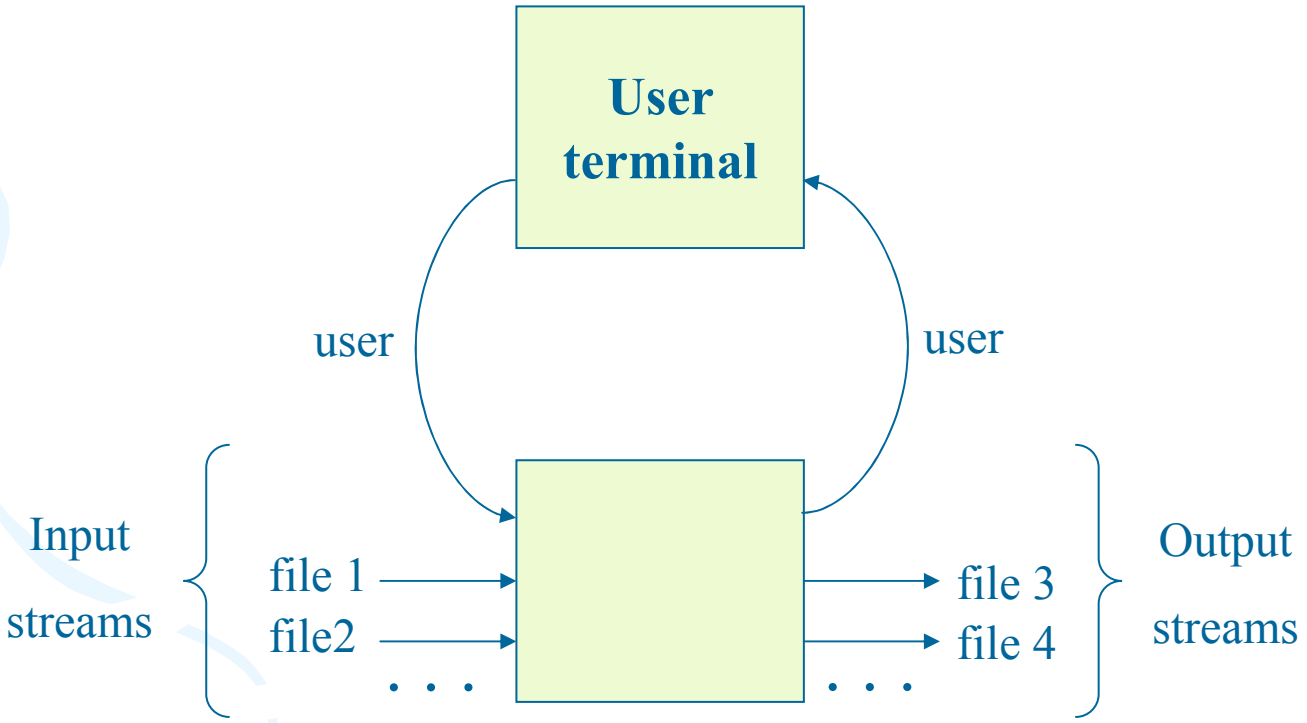
razavi@Comp.iust.ac.ir

۱۳۸۳

مقدمه

- برقراری ارتباط با فایل ها
- پردازش فایل های حاوی عبارات
- کار با کاراکترها
- ایجاد و تجزیه اتم ها

برقراری ارتباط با فایل ها



برقراری ارتباط با فایل ها

- در هر زمان در طول اجرای یک برنامه پرولوگ تنها دو فایل فعال هستند: یکی برای ورودی و دیگری برای خروجی
  - جریان ورودی فعلی (current input stream)
  - جریان خروجی فعلی(current output stream)
- در ابتدای اجرا این دو جریان مرتبط با جریان user می باشند.
- می توان بوسیله رابطه زیر جریان ورودی فعلی را تغییر داد:

see(Filename)

یک مثال نوعی برای استفاده از see:

```
…
see( file1),
read_from_file( Information),
see( user),
…
```

برقراری ارتباط با فایل ها

- می توان جریان خروجی فعلی را با یک هدف مانند زیر تغییر داد:

tell(Filename)

یک مثال نوعی برای استفاده از tell:

```
...
tell( file2),
write_on_file( file2),
see( user),
```

هدف seen جریان ورودی فعلی و هدف told جریان خروجی فعلی را می بندد.

پردازش فایل های حاوی عبارات

- نحوه خواندن و نوشتن در فایلها
  - کاراکتر به کاراکتر (get, get0, put).
  - عبارت به عبارت (read, write).
- read :
  - مسند read برای خواندن یک عبارت از جریان ورودی فعلی بکار می رود
  - مسند read(X)، باعث می شود عبارت بعدی T خوانده شده و این عبارت با X تطابق یابد.
  - در هنگام رسیدن به انتهای جریان ورودی فعلی، X با مقدار اتم end_of_file نمونه دهی می شود.

پردازش فایل های حاوی عبارات

- write:
  - هدف write(X) عبارت X را به جریان خروجی فعلی می فرستد.
- برای قالب بندی خروجی می توان از مسند پیش ساخته tab(N) برای درج فاصله به تعداد N و از nl برای درج یک خط جدید در خروجی استفاده نمود.

پردازش فایل های حاوی عبارات

- مثال:

```
cube :-
    read( X),
    process( X).

process( stop) :- !.
process( N) :-
    C is N * N * N,
    write( X),
    cube.
```

```
?- cube.
2.
8
5.
125
12
1728
stop.
yes
```

پردازش فایل های حاوی عبارات

- شکل بهتر برای مثال قبل:

```
cube :-
    write( 'Next item, please:'),
    read( X),
    process( X).

process( stop) :- !.
process( N) :-
    C is N * N * N,
    write( 'Cube of  '), write( N),
    write( 'is '), write( C), nl,
    cube.
```

```
?- cube.
Next item, please: 5.
Cube of 5 is 125
Next item, please: 12.
Cube of 12 is 1728
Next item, please: stop.
yes
```

پردازش فایل های حاوی عبارات

- نمایش لیست ها

```
writelist( []).
writelist( [H| T] ) :-
    write( H), nl,
    writelist( T).
```

```
?- writelist( [a, b, c] ).

a

b

c
```

پردازش فایل های حاوی عبارات

- یک مثال دیگر:

```
bars( []).
bars( [N| L] ) :-
    stars( N), nl,
    bars( L).

stars( N) :-
    N > 0,
    write( '*'),
    N1 is N -1,
    stars( N1).
stars( N) :-
    N =< 0.
```

```
?- bars( [3,4,6,5] ).
***

****

******

*****
```

پردازش فایل های حاوی عبارات

- یک مثال از پردازش فایل حاوی عبارت ها:

…, see(F), processfile, see(user),…

```
processfile :-
    read( Term),
    process( Term).

process( end_of_file) :- !.
process( Term) :-
    treat( Term),
    processfile.
```

کار با کاراکترها

- نوشتن یک کاراکتر در جریان خروجی فعلی:

put(C)

که در آن C برابر کد اسکی کاراکتری است که می خواهیم در خروجی بنویسیم.

مثال:

?- put(65), put (66), put (67).

ABC

کار با کاراکترها

- خواندن یک کاراکتر از جریان ورودی فعلی با هدف زیر انجام می شود:

get0(C)

نوع دیگر یعنی get برای خواندن کاراکترهای قابل چاپ بکار می رود. از روی تمام کاراکترهای غیرقابل چاپ پرش می کند.

کار با کاراکترها

- مثال: خواندن یک جمله و حذف فاصله های اضافی در آن:

```
squeeze :-
    get0( C),
    put( C),
    dorest( C).
dorest( 46) :- !.      % full stop
dorest( 32) :- !,      % blank
    get( C),
    put( C),
    dorest( C).
dorest( Letter) :-
    squeeze.
```

```
?- squeeze.
This    is   a        test.
This is a test
yes
```

ایجاد و تجزیه اتم ها

- مسند name(A, L):اتم ها را بانمایش اسکی آنها مربوط می سازد.

مثال: مقدار عبارت زیر درست می باشد.

name(zx32 , [122, 120, 50 51 50]).

کاربرد های name:

(۱) تجزیه یک اتم به کاراکترهایش

(۲) ترکیب لیستی از کاراکترها در یک اتم

ایجاد و تجزیه اتم ها

- مثال از کاربرد اول:

در یک برنامه حمل و نقل که با سفارشها، تاکسی ها و رانندگان مانند order1, order2, taxi1, taxilux, driver1 سروکار دارد مسند taxi(X) چک می کند که آیا اتم X بیانگر یک تاکس می باشد یا خیر:

```
taxi( X) :-
    name( X, XList),
    name( taxi, TList),
    conc( TList, _, XList).  % is word ‘taxi’ is prefix of X
```

ایجاد و تجزیه اتم ها

- مثال از کاربرد دوم: تبدیل یک جمله زبان طبیعی به لیستی از کلمات تشکیل دهنده آن :

مثال:

This is a test.

getsentence(Sentence).

Sentence = [‘This’, is, a, test]

ایجاد و تجزیه اتم ها

- برنامه مثال قبل

```
getsentence( WordList) :-
      get0( Char),
      getrest( Char, WordList).
getrest( 46, []) :-      !.
getrest( 32, WordList) :-   - !,
      getsentence( WordList).
getrest( Letter, [Word| WordList] ) :-
      getletters( Letter, Letters, Nextchar),
      name( Word, Letters),
      getrest( Nextchar, WordList).
getletters( 46, [] , 46) :-  - !.
 getletters( 32, [], 32) :-  - !.
getletters( Let, [Let| Letters], Nextchar) :-
      get0( Char),
      getletters( Char, Letters, Nextchar).
```