File을 다루는 명령들

IDL

시작하며

일반적으로 운영체제에서 다루는 파일 찾기, 복사, 삭제 등의 기능을 구현한 IDL 함수와 프로시저를 소개합니다. 별표의 개수는 유용성 또는 사용빈도를 의미하며 주관적입니다. 각 명령어 마다 미처 소개하지 못한 키워드가 더 있습니다. 여기에 설명된 기능이 조금 아쉽게 느껴진다면 도움말을 한번 검토해 보세요. 필요한 키워드가 보일 것입니다.

CD ***

지정 디렉토리로 이동하거나, CURRENT 키워드를 이용하여 현재 디렉토리 위치를 확인할 때 사용

CD [, '이동하고자 하는 곳'] [, CURRENT=변수]

```
IDL> cd, ´c:\´  ;Windows 계열의 위치 이동
IDL> cd, ´/usr/local´          ;Unix 계열의 위치 이동
IDL> cd, current=nowpos     ;현재 위치 확인
IDL> print, nowpos
c:\
```

FILEPATH ***

IDL 설치 디렉토리 기준으로 경로지정. SUBDIR 키워드를 이용하여 하위 디렉토리 지정.

Result=FilePath('파일명' [,SUBDIR=하위디렉토리배열])

```
IDL> print, filepath(´head.dat´, subdir=[´examples´, ´data´])
C:\Program Files\ITT\IDL\IDL81\examples\data\head.dat
```

PATH_SEP **

운영체제에 따른 디렉토리 구분자를 리턴. Windows는 역슬래시(\), Unix 계열은 슬래시(/).

Result=PATH_SEP()

```
IDL> libdir=!dir+path_sep()+´lib´   ;!DIR은 IDL 설치 디렉토리
IDL> print, libdir
C:\Program Files\ITT\IDL\IDL81\lib          ;Windows
/usr/local/itt/idl/idl81/lib                ;Unix 계열
```

FILE_WHICH **

지정경로, 또는 IDL의 검색 경로(!PATH 즉, 환경설정의 경로에 해당)를 검색하여 지정파일을 찾고 존재할 경우 전체 경로(Full Name) 리턴.

Result=File_Which([지정경로,] '파일명')

키워드 : /INCLUDE_CURRENT_DIR

```
IDL> print, file_which(´head.dat´)
C:\Program Files\ITT\IDL\IDL81\examples\data\head.dat
IDL> print, file_which(´readcol.pro´)
C:\Users\yi\IDLAddon\astron\pro\readcol.pro
```

FILE_MKDIR ***

디렉토리 생성.

FILE_MKDIR, '생성DIR이름1' [, '생성DIR이름2', ...]

```
IDL> file_mkdir, ´C:\myidltest´
```

FILE_SEARCH *****

파일 시스템에서 지정 패턴에 매칭이 되는 모든 파일을 검색하여 리턴함. 다양한 키워드로 파일의 조건을 제한할 수 있음.

Result=File_Search('파일패턴 ') ;(또는)

Result=File_Search('디렉토리패턴 ', '재귀패턴')

키워드 : Count = 변수, /FOLD_CASE, /NOSORT, /TEST_DIRECTORY, /TEST_EXECUTABLE, /TEST_READ, /TEST_REGULAR, /TEST_WRITE, /TEST_ZERO_LENGTH, /FULLY_QUALIFY_PATH

```
IDL> cd, ´C:\Windows´
IDL> files=file_search()   ;현재디렉토리의 모든 파일 검색
IDL> print, files, format=´(A)´

IDL> files=file_search(/FULLY_QUALIFY_PATH)  ;전체경로포함

IDL> cd, filepath(´lib´)  ;IDL 설치디렉토리 아래 lib 디렉토리로 이동
IDL> files=file_search(´a*.pro´)   ;a로 시작하는 .pro 파일 검색
IDL> print, files, format=´(A)´
a_correlate.pro
adapt_hist_equal.pro
.....

;IDL의 examples 디렉토리 아래 모든 서브 디렉토리를 검색.
;디렉토리만 리턴함(/test_directory 키워드).
IDL> dirs=file_search(filepath(´examples´), ´*´, /test_directory)
IDL> print, dirs, format=´(A)´

;하위의 모든 서브디렉토리에서 .pro 파일을 검색
;검색된 파일의 개수는 t 변수에 저장
IDL> files=file_search(!dir, ´*.pro´, count=t)
```

온라인도움말을 보면, 재미있는 예제가 더 있습니다. FILE_SEARCH() 함수의 다양한 기능을 확인해 보세요.

FILE_COPY ***

파일을 복사함. *나 ? 같은 와일드 카드를 사용할 수 있으며 상위디렉토리를 의미하는 점 두 개('..')나 현재 디렉토리를 의미하는 점('.')도 사용 가능.

FILE_COPY, '원본', '복사경로'

키워드 : /ALLOW_SAME, /OVERWRITE, /RECURSIVE

```
IDL> cd, ´C:\myidltest´  ;test를 위한 디렉토리로 이동
IDL> libdir=!dir+path_sep()+´lib´+path_sep()
;lib 디렉토리의 모든 .pro 파일을 현재 디렉토리로 복사
IDL> file_copy, libdir+´*.pro´, ´.´
;하위디렉토리를 모두 포함하는 파일을 복사. 덮어쓰기 허용
IDL> file_copy, libdir+´*´, ´.´, /recursive, /overwrite
```

FILE_MOVE **

파일을 다른 위치로 옮기거나 이름을 바꿀 때 사용.

FILE_MOVE, '원본', '새이름 또는 옮겨질 파일경로'

키워드 : /ALLOW_SAME, /OVERWRITE

```
IDL> file_move, ´a_correlate.pro´, ´acorrel.pro´  ;이름변경
IDL> file_move, ´acorrel.pro´, ´obsolete´ ;obsolete DIR로 이동
```

FILE_DELETE ***

지정된 파일을 삭제.(실습 시에 주의하세요!!!)

FILE_DELETE, '파일1', '파일2', ... ;(또는)

FILE_DELETE, 삭제파일목록배열

키워드 : /ALLOW_NONEXISTENT, /RECURSIVE

```
IDL> files=file_search(´C:\myidltest\´, ´*´)
;하위 디렉토리 포함 모두 삭제. 파일 존재하지 않더라도 허용
IDL> file_delete, files, /recursive, /allow_nonexistent
```

FILE_TEST ****

파일 존재여부 확인. 키워드를 이용하여 다양한 테스트.

Result=File_Test('파일경로')

키워드 : /EXECUTABLE, /READ, /WRITE, /ZERO_LENGTH, /REGULAR, /DIRECTORY, GET_MODE=변수

```
IDL> file1=filepath(´mars.jpg´, subdir=[´examples´, ´data´])
IDL> print, file_test(file1)
           0                    ;존재하지 않는 파일
IDL> file2=filepath(´marsglobe.jpg´, subdir=[´examples´, ´data´])
IDL> print, file_test(file2)
           1                    ;존재하는 파일

IDL> test=file_test(file2, get_mode=mode)
IDL> print, mode, format=´(o)´   ;8진수 형태로 출력
         666                    ;rw-rw-rw- 의미
```

FILE_CHMOD **

파일의 읽기,쓰기,실행 권한 설정. Windows의 경우 사용자, 그룹, 전체의 구분이 명확하지 않음.

FILE_CHMOD, 파일 또는 파일배열 [,모드]

키워드 : /A_READ, /A_WRITE, /A_EXECUTE (for All)
/G_READ, /G_WRITE, /G_EXECUTE (for Group)
/U_READ, /U_WRITE, /U_EXECUTE (for User)

```
IDL> file_chmod, myfile, ´644´o      ;8진 표현 rw-r--r-- 의미
IDL> file_chmod, myfile, G_WRITE=1  ;그룹의 쓰기권한 허용
```

FILE_LINK 와 FILE_READLINK *

UNIX 계열에서만 사용할 수 있습니다. 하드 링크 또는 심볼릭 링크를 설정하고 조회할 수 있는 기능입니다. 필요시 도움말을 확인하세요.

FILE_INFO ****

파일에 대한 다양한 정보를 획득. 파일 크기, 생성일자 등을 포함하여 FILE_TEST에서 점검할 수 있는 내용도 확인 가능. 리턴값은 구조체로 파일에 관한 모든 정보가 저장됨.

Result=FILE_INFO('파일')

```
IDL> file=filepath(´head.dat´, subdir=[´examples´, ´data´])
IDL> info=file_info(file)
IDL> help, info, /struct
** Structure FILE_INFO, 21 tags, length=72, data length=67:
   NAME   STRING  ´C:\Program Files\ITT\...\head.dat´ (이름)
   EXISTS      BYTE    1    (존재여부 FILE_TEST결과)
   READ        BYTE    1    (읽기권한)
   WRITE       BYTE    1    (쓰기권한)
   EXECUTE     BYTE    0    (실행권한)
   REGULAR         BYTE    1    (일반파일인가?)
   DIRECTORY       BYTE    0    (디렉토리인가?)
   BLOCK_SPECIAL BYTE   0
                    (UNIX Block Special Device인가?)
   CHARACTER_SPECIAL   BYTE   0
                    (UNIX Character Special Device 인가?)
   NAMED_PIPE    BYTE       0
                    (UNIX Named PIPE인가?)
   SETUID    BYTE    0          (UNIX의 Set-User-ID bit set)
   SETGID    BYTE    0          (UNIX의 Set_group-ID bit set)
   SOCKET    BYTE    0          (UNIX Domain Socket)
   STICKY_BIT  BYTE     0       (UNIX Sticky Bit set)
   SYMLINK     BYTE     0       (UNIX 심볼릭 링크 여부)
   DANGLING_SYMLINK   BYTE   0
          (존재하지 않는 파일에 대한 UNIX 심볼릭 링크인지?)
   MODE    LONG    438         (파일권한Mode)
   ATIME   LONG64  1312597396 (최종 접근 시각)
   CTIME   LONG64  1299732956 (생성 또는 최종 상태변경시각)
   MTIME   LONG64  1299732956 (최종 수정 시각)
   SIZE    LONG64  456000      (파일 크기, byte)
```

FILE_DIRNAME ***

주어진 파일 경로에서 디렉토리부분을 리턴. 파일경로는 배열도 가능.

Result=FILE_DIRNAME('파일경로')

키워드 : /MARK_DIRECTORY

```
IDL> file=file_which(´elev_t.jpg´)
IDL> print, file
C:\Program Files\ITT\IDL\IDL81\examples\data\elev_t.jpg
IDL> print, file_dirname(file)
C:\Program Files\ITT\IDL\IDL81\examples\data
IDL> print, file_dirname(file, /MARK_DIRECTORY)
C:\Program Files\ITT\IDL\IDL81\examples\data\
```

/MARK_DIRECTORY 키워드를 사용하면 리턴값의 맨 끝에 PATH_SEP() 의 결과를 붙이는 것과 같은 결과가 됩니다.

FILE_BASENAME ****

주어진 파일 경로에서 파일이름(디렉토리들을 제외한 맨 뒷부분)을 리턴. 파일 경로는 배열도 가능.

Result=FILE_BASENAME('파일경로' [,'제거대상확장자'])

```
IDL> print, file
C:\Program Files\ITT\IDL\IDL81\examples\data\elev_t.jpg
IDL> print, file_basename(file)
elev_t.jpg
IDL> print, file_basename(file, ´.jpg´)
elev_t
```

제거대상확장자와 파일의 맨 끝부분 일치되는 범위를 제거합니다.