Published: 09/23/68

Identification

iocall - i/o system interface command.
Ken Thompson

Introduction

Often it is useful to issue i/o system outer calls from command level. The command iocall is provided for this purpose. iocall will perform the following functions: 1) It will accept a variety of argument formats, supplying useful default arguments where required; 2) it will print values of return arguments; 3) it will decode and print status returned by the i/o system.

Usage

iocall is a command designed to be called by the shell. The generic form of the iocall command is :

```
iocall outercall ioname arguments
```

Outercall is one of the i/o system outer calls "attach", "detach", "read", "write", "seek", "tell", "setsize", or "getsize". As more i/o outer calls are implemented, iocall will be updated to include them. Ioname is the ioname argument present on all outer calls (See BF.1.00).

iocall supplies a status argument on all i/o calls it makes. On return from the i/o system, this status argument is decoded and comments are typed on the stream "user_output". Since other modules making outer calls may wish to have status checked, the following entry in iocall is provided for this purpose.

```
call iocall$checkstatus(status);
        dcl status bit(72);
```

The status returned by the i/o system is described in MSPM BF.1.07.

Outer Calls

Below is a list of the outer calls accepted by iocall. The list starts with the complete i/o system outer call. Following the outer call is a list of the variations of the call acceptable by iocall. Following this are notes on special cases associated with the outer call.

1) attach

```
call attach(ioname, type, ioname2, mode, status);
```

```
iocall attach ioname type ioname2
iocall attach ioname type ioname2 mode
```

If the mode argument is missing, the null character string is supplied. The status argument is supplied and decoded.

2) detach

```
call detach(ioname, ioname2, mode, status);

iocall detach ioname
iocall detach ioname ioname2
iocall detach ioname ioname2 mode
```

If the arguments ioname2 or mode are missing, the null character string is supplied. The status argument is supplied and decoded.

3) read

```
call read(ioname, workspace, offset, nelem, nelemt, status);

iocall read ioname segment
iocall read ioname segment nelem
iocall read ioname segment offset nelem
```

Offset and nelem arguments, if present, are in decimal. If the offset argument is missing, 0 is supplied. If the nelem argument is missing, the maximum size of segment is provided as a multiple of the current element size. A pointer to the base of segment is supplied as the workspace argument. If segment does not exist, it is created in the working directory. The status argument is supplied and decoded.

4) write

```
call write(ioname, workspace, offset, nelem, nelemt, status);

iocall write ioname segment
iocall write ioname segment nelem
iocall write ioname segment offset nelem
```

Offset and nelem arguments, if present, are in decimal. If the offset argument is missing, 0 is supplied. If the nelem argument is missing, the bit count of segment is provided as a multiple of the current element size. A pointer to the base of segment is supplied as the workspace argument. The status argument is supplied and decoded.

5) seek

```
call seek(ioname, ptrname1, ptrname2, offset, status);

iocall seek ioname ptrname1
```

```
iocall seek ioname ptrname1 ptrname2
iocall seek ioname ptrname1 ptrname2 offset
```

Offset, if present, is decimal. If offset is missing, 0 is supplied. If ptrname2 is missing, "first" is supplied. The status argument is supplied and decoded.

6) tell

```
call tell(ioname, ptrname1, ptrname2, offset, status);

iocall tell ioname ptrname1
iocall tell ioname ptrname1 ptrname2
```

If ptrname2 is missing, "first" is supplied. The offset argument is supplied and its value is printed in decimal on return. The status argument is supplied and decoded.

7) setsize

```
call setsize(ioname, elementsize, status);

iocall setsize ioname elementsize
```

The elementsize argument is decimal. The status argument is supplied and decoded.

8) getsize

```
call getsize(ioname, elementsize, status);

iocall getsize ioname
```

The elementsize argument is provided and its value is printed in decimal on return. The status argument is supplied and decoded.