NAME

aread – asynchronous read from a device

SYNOPSIS

#include <sys/uio.h>
#include <sys/aio_req.h>
#include <sys/cred.h>
#include <sys/ddi.h>
#include <sys/sunddi.h>

int *prefix***aread(dev_t** *dev***, struct aio_req** **aio_reqp***, cred_t** **cred_p***);**

ARGUMENTS

dev Device number.

aio_reqp Pointer to the **aio_req**(9S) structure that describes where the data is to be stored.

cred_p Pointer to the credential structure.

INTERFACE LEVEL

Solaris DDI specific (Solaris DDI). This entry point is **Optional**. Drivers that do not support an **aread**() entry point should use **nodev**(9F)**.**

DESCRIPTION

The driver's **aread**( ) routine is called to perform an asynchronous read. **getminor**(9F) can be used to access the minor number component of the *dev* argument. **aread**( ) may use the credential structure pointed to by *cred_p* to check for superuser access by calling **drv_priv**(9F). The **aread**( ) routine may also examine the **uio**(9S) structure through the **aio_req** structure pointer, *aio_reqp*. However, the **aread**( ) routine must not modify *aio_reqp* or the contents of the **aio_req**(9S) structure. **aread**( ) must call **aphysio**(9F) with the **aio_req** pointer, and a pointer to the driver's **strategy**(9E) routine.

RETURN VALUES

The **aread**( ) routine should return **0** for success, or the appropriate error number.

CONTEXT

This function is called from user context only.

EXAMPLES

The following is an example of an **aread**( ) routine.

```
static int
xxaread(dev_t dev, struct aio_req *aio, cred_t *cred_p)
{
        int instance;
        struct xxstate *xsp;

        instance = getminor(dev);
        xsp = ddi_get_soft_state(statep, instance);

        /*Verify soft state structure has been allocated */
        if (xsp == NULL)
                return (ENXIO);

        return (aphysio(xxstrategy, anocancel, dev, B_READ, xxminphys, aio));
}
```

SEE ALSO

read(2), **aioread**(3), **awrite**(9E), **read**(9E), **strategy**(9E), **write**(9E), **anocancel**(9F), **aphysio**(9F), **ddi_get_soft_state**(9F), **drv_priv**(9F), **getminor**(9F), **minphys**(9F), **aio_req**(9S), **cb_ops**(9S), **uio**(9S)

Writing Device Drivers

BUGS

There is no way other than calling **aphysio**(9F) to accomplish an asynchronous read.