

```

algorithm vfault      /* handler for validity faults */
input:  address where process faulted
output: none
{
    find region, page table entry, disk block descriptor
        corresponding to faulted address, lock region;
    if (address outside virtual address space)
    {
        send signal (SIGSEGV: segmentation violation) to process;
        goto out;
    }
    if (address now valid)      /* process may have slept above */
        goto out;
    if (page in cache)
    {
        remove page from cache;
        adjust page table entry;
        while (page contents not valid) /* another proc faulted first */
            sleep (event contents become valid);
    }
    else      /* page not in cache */
    {
        assign new page to region;

        put new page in cache, update pfddata entry;
        if (page not previously loaded and page "demand zero")
            clear assigned page to 0;
        else
        {
            read virtual page from swap dev or exec file;
            sleep (event I/O done);
        }
        awaken processes (event page contents valid);
    }
    set page valid bit;
    clear page modify bit, page age;
    recalculate process priority;
out:  unlock region;
}

```

Figure 9.21. Algorithm for Validity Fault Handle.