

NAME

namecache, **cache_lookup**, **cache_revlookup**, **cache_enter**, **cache_purge**,
cache_purgevfs, **namecache_print** – name lookup cache

SYNOPSIS

```
#include <sys/namei.h>
#include <sys/proc.h>
#include <sys/uio.h>
#include <sys/vnode.h>

int
cache_lookup(struct vnode *dvp, struct vnode **vpp,
             struct componentname *cnp);

int
cache_revlookup(struct vnode *vp, struct vnode *dvp, char **bpb, char *bufp);

void
cache_enter(struct vnode *dvp, struct vnode *vp, struct componentname *cnp);

void
cache_purge(struct vnode *vp);

void
cache_purgevfs(struct mount *mp);

void
namecache_print(struct vnode *vp, void (*func)(const char *, ...));
```

DESCRIPTION

The name lookup cache is the mechanism to allow the file system type dependent algorithms to quickly resolve a file's vnode from its pathname. The name lookup cache is generally accessed through the higher-level namei(9) interface.

The name of the file is used to lookup an entry associated with the file in the name lookup cache. If no entry is found, one is created for it. If an entry is found, the information obtained from the cache lookup contains information about the file which is useful to the file system type dependent functions.

The name lookup cache is managed by a least recently used (LRU) algorithm so frequently used names will hang around. The cache itself is a hash table called *nchashtbl*, containing *namecache* entries that are allocated dynamically from a kernel memory pool. Each entry has the following structure:

```
#define NCHNAMLEN      31      /* maximum name segment length */
struct namecache {
    LIST_ENTRY(namecache) nc_hash; /* hash chain */
    TAILQ_ENTRY(namecache) nc_lru; /* LRU chain */
    LIST_ENTRY(namecache) nc_vhash; /* directory hash chain */
    struct vnode *nc_dvp; /* vnode of parent of name */
    u_long nc_dvpid; /* capability number of nc_dvp */
    struct vnode *nc_vp; /* vnode the name refers to */
    u_long nc_vpid; /* capability number of nc_vp */
    char nc_nlen; /* length of name */
    char nc_name[NCHNAMLEN]; /* segment name */
};
```

For simplicity (and economy of storage), names longer than a maximum length of NCHNAMLEN are not cached; they occur infrequently in any case, and are almost never of interest.

Each *namecache* entry can appear on two hash chains in addition to *nshashtbl*: *ncvhashtbl* (the name cache directory hash chain), and *nclruhead* (the name cache LRU chain). The hash chains are indexed by a hash value obtained from the file's name and the address of its parent directory vnode.

Functions which access to the name cache pass arguments in the partially initialised *componentname* structure. See *vnodeops(9)* for details on this structure.

FUNCTIONS

cache_lookup(*dvp*, *vpp*, *cnp*)

Look for a name in the cache. **cache_lookup()** is called with *dvp* pointing to the vnode of the directory to search and *cnp* pointing to the partially initialised component structure. *cnp->cn_nameptr* points to the name of the entry being sought, *cnp->cn_name_len* tells the length of the name, and *cnp->cn_hash* contains a hash of the name. If the lookup succeeds, the vnode is locked, stored in *vpp* and a status of zero is returned. If the locking fails for whatever reason, the vnode is unlocked and the error is returned. If the lookup determines that the name does not exist any longer, a status of ENOENT is returned. If the lookup fails, a status of -1 is returned.

cache_revlookup(*vp*, *dvp*, *bpp*, *bufp*)

Scan cache looking for name of directory entry pointing at *vp* and fill in *dvp*. If *bufp* is non-NULL, also place the name in the buffer which starts at *bufp*, immediately before *bpp*, and move *bpp* backwards to point at the start of it. Returns 0 on success, -1 on cache miss, positive *errno* on failure.

cache_enter(*dvp*, *vp*, *cnp*)

Add an entry to the cache. **cache_enter()** is called with *dvp* pointing to the vnode of the directory to enter and *cnp* pointing to the partially initialised component structure. *cnp->cn_nameptr* points to the name of the entry being entered, *cnp->cn_name_len* tells the length of the name, and *cnp->cn_hash* contains a hash of the name.

cache_purge(*vp*)

Flush the cache of a particular vnode *vp*. **cache_purge()** is called when a vnode is renamed to hide entries that would now be invalid.

cache_purgevfs(*mp*)

Flush cache of a whole file system *mp*. **cache_purgevfs()** is called when file system is unmounted to remove entries that would now be invalid.

namecache_print(*vp*, *func*)

Print all entries of the name cache. *func* is the *printf(9)* format. **namecache_print()** is only defined if the kernel option DDB is compiled into the kernel.

CODE REFERENCES

This section describes places within the NetBSD source tree where actual code implementing or using the name lookup cache can be found. All pathnames are relative to */usr/src*.

The name lookup cache is implemented within the file *sys/kern/vfs_cache.c*.

SEE ALSO

intro(9), *namei(9)*, *vfs(9)*, *vnode(9)*

HISTORY

The name lookup cache first appeared in 4.2BSD.

AUTHORS

The original name lookup cache was written by Robert Elz.