

The Paged File (PF) Layer: System Report

Rut Diane Cuebas (2019-28748)

Jungwoo Yang (2019-23417)

1. File Structure

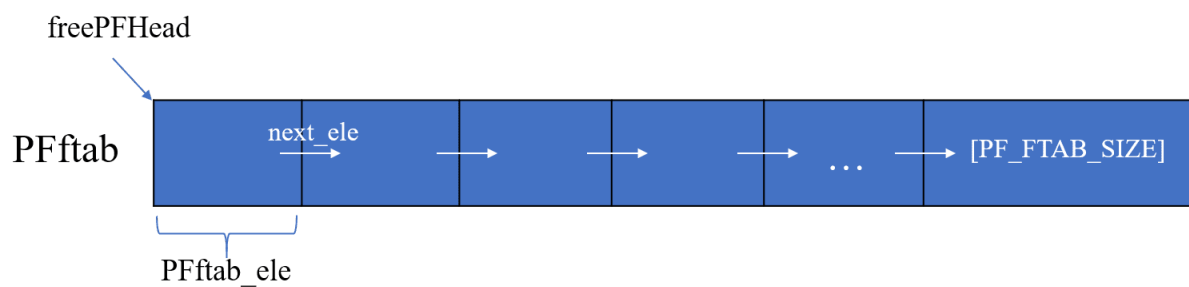
pf/pf.c - Contains implementations for pf.h

h/pf.h - Contains function prototypes that are implemented in pf.c

pf/pfinternal.c – Contains implementations for pfinternal.h and struct definitions for the PF layer's file table entries, hash entries, and the file page header.

pf/pfinternal.h – Contains prototypes that are implemented in pfinternal.c

2. Data Structures



PFftab

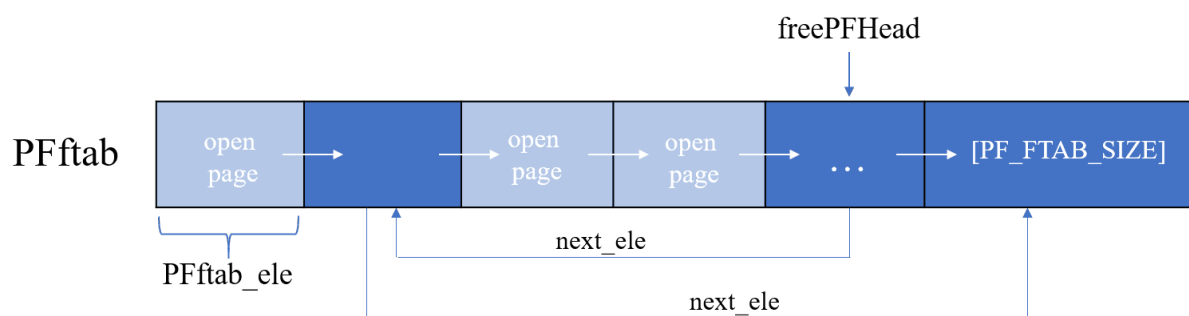
The PF File table. It is a linked list that consists of entries of type **PFftab_ele**. Each entry points to the next entry with a pointer called **next_ele**. The file table size is **PF_FTAB_SIZE**.

freePFHead

A pointer to the head of a list of empty (invalid) PF File table entries. Like the **PFftab** it has a **next_ele** pointer which points to the next element. When the PF layer is initialized it points to PFftab[0].

As entries in the file table become valid then freePFHead is moved to where its **next_ele** is pointing. As entries become invalid **freePFHead** points to the recently invalidated entry, and the **next_ele** pointer points to the previous head.

Takes O(1) time to insert or remove an element.

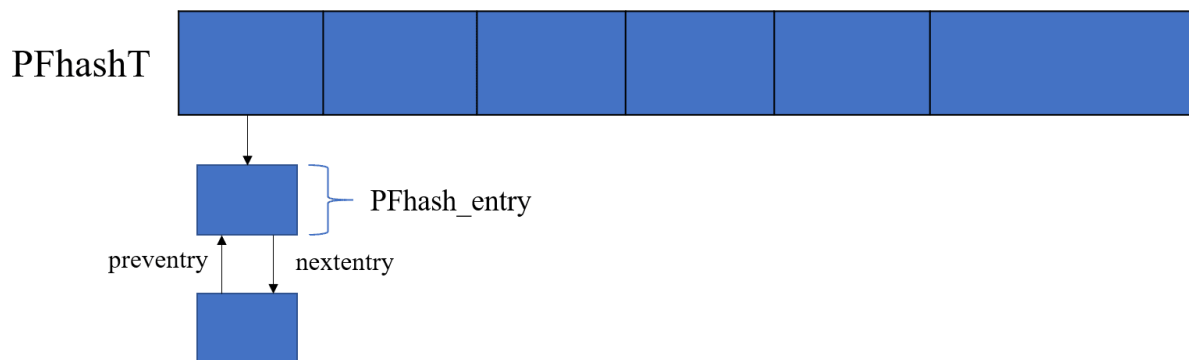


PFhdr_str: struct for each file's header.

```
1. typedef struct PFhdr_str {
2.     int    numpages;    /* number of pages in the file */
3. } PFhdr_str;
```

PFftab_ele: struct for the PF File table elements.

```
1. typedef struct PFftab_ele {
2.     bool_t    valid;    /* set to TRUE when a file is open. */
3.     ino_t     inode;    /* inode number of the file */
4.     char      fname[50]; /* file name */
5.     int       unixfd;    /* Unix file descriptor */
6.     PFhdr_str hdr;      /* file header */
7.     short     hdrchanged; /* TRUE if file header has changed */
8.
9.     struct PFftab_ele *next_ele;
10. } PFftab_ele;
```



PFhashT

The PF hash table uses filename to find out the index of the file (PFhashF). Consists of doubly linked entries of type **PFhash_entry**.

The hash table is initialized as **PFfreeHash[MAXOPENFILES]** to get rid of unnecessary memory allocation.

This can speed up finding files in the **PFftab_ele** by filename

PFHash_entry: struct for the PF layer's hash table entries.

```
1. typedef struct PFHash_entry{
2.     struct PFHash_entry *nextentry;
3.     struct PFHash_entry *preentry;
4.     struct PFftab_ele *PF_ele;
5. } PFHash_entry;
```