

Active Module Statistics - Design and Code Document

Problem statement

Display the device driver/module statistics currently active in the Operating System.

Solution

On closer inspection of the `modules` file in `/proc`, we note that the following details are specified. - name - size - instances - dependencies - status - offset

Apart from the above details, we run the command line utility, `modinfo` on each of the modules to obtain more details. - filename - license - description - author(s) - srcversion - vermagic - sig_key - sig_hashalgo - parm(s)

Note: This list is non-exhaustive and depends on the module.

The outputs of the two above mentioned utilities are concatenated and output for each of the active modules. Active modules are determined by the number of instances currently running.

Data Structures

This is the structure that stores data from the `/proc/modules` file.

```
typedef struct my_modules
{
    char name[256];
    unsigned int size;
    unsigned int instances;
    unsigned int number_of_dependencies;
    char dependencies[500];
    unsigned int status;
    unsigned long int offset;
} Module;
```

Functions

Refer to the header files for elaborate comments on the functionality of each of the following functions.

```
void module_initializer(  
    Module *module,  
    char *name,  
    unsigned int size,  
    unsigned int instances,  
    unsigned int number_of_dependencies,  
    char *dependencies,  
    unsigned int status,  
    unsigned long int offset  
);  
void module_reader();  
int parser(Module *module, char *line);  
int linecounter();  
void modinfo(char *module);
```

Specifications

- Operating System: 18.04.1-Ubuntu (Linux version 5.0.0-29-generic)
- Processor(s): Intel® Core™ i5-7200U CPU @ 2.50GHz × 4