1. System call

The system call that we are going to implement is chown(). Chown() is used to change the owner and groking of the world blackbeardup of a file or a directory. To use it we must include the **unistd.h** library. The syntax is : chown(*pathname, owner_id, group_id);

- Parameters:
  - ✓ pathname: The path to the file or directory whose ownership is to be changed. It's a **pointer** variable with the data type **char**.
  - ✓ owner_id: The user ID of the new owner. Using -1 to keeps the current owner. Its data type is **uid_t**.
  - ✓ Group_id: The group ID of the new group. Using -1 to keeps the current group. Its data type is **gid_t**.

- Return Values:
  - ✓ It returns **0** on success.
  - ✓ It returns **-1** on failure
  - ✓ It returns **errno** is set to indicate the error.

❖ Now we will see how it works in Minix.

1. First we will create a c file called trial.c using nano.

Image 2. Creating a c file

2. Then we will create a text file and implement **chown()** using the c programming language.

➢ The code and the minix implementation are given below.

```c
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <errno.h>

int main() {
const char *fn = "chown.txt";
mode_t m = S_IRUSR | S_IWUSR | S_IRGRP | S_IROTH;
int c = creat(fn, m);
if (c == -1) {
perror("Error creating the file!");
return 1;
}

printf("File '%s' created successfully with mode %o.\n", fn, m);

uid_t no = 1002;
gid_t ng= 1001;

if (chown(fn, no, ng) == -1) {
perror("Error changing the ownership!");
close(c);
return 1;
}

printf("Ownership of '%s' changed successfully to UID: %d and
GID: %d.\n", fn, no, ng);

if (close(c) == -1) {
perror("Error closing the file!");
return 1;
}

printf("File '%s' closed successfully.\n", fn);

return 0;
}
```

```
  GNU nano 2.2.6                File: trial.c

#include<stdio.h>
#include<fcntl.h>
#include<unistd.h>
#include<sys/stat.h>
#include<sys/types.h>
#include<errno.h>

int main(){

const char *fn = "chown.txt";
mode_t m = S_IRUSR|S_IWUSR|S_IRGRP|S_IROTH;
int c = creat(fn,m);


if(c==-1){
perror("Error creating the file!");
return 1;
}

printf("File '%s' created successfully with mode %o.\n",fn,m);
                        [ Read 41 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

Image 3. creating a .txt file called chown.txt

```
  GNU nano 2.2.6                File: trial.c

int c = creat(fn,m);


if(c==-1){
perror("Error creating the file!");
return 1;
}

printf("File '%s' created successfully with mode %o.\n",fn,m);

uid_t no = 1002;
gid_t ng = 1001;

if(chown(fn,no,ng)==-1){
perror("Error changing the ownership!");
close(c);
return 1;
}

printf("Ownership of '%s' changed successfully to UID: %d and GID: %d.\n",fn,no$

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

Image 4. setting up user and group IDs

- ✧ We will use the IDs of the user and group that we created earlier. We can get the IDs by using the **chfn** command.

```
  GNU nano 2.2.6                File: trial.c

gid_t ng = 1001;

if(chown(fn,no,ng)==-1){
perror("Error changing the ownership!");
close(c);
return 1;
}

printf("Ownership of '%s' changed successfully to UID: %d and GID: %d.\n",fn,no$

if(close(c)==-1){
perror("Error closing the file!");
return 1;
}

printf("File '%s' closed successfully.",fn);

return 0;
}

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

Image 5. Implementing chown() and closing the file

```
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
# clang -o s trial.c
# _
```

Image 6. Compiling the code and creating an executable file called “s”

✧ The code has been compiled with out errors.

Image 7. Running the executable file

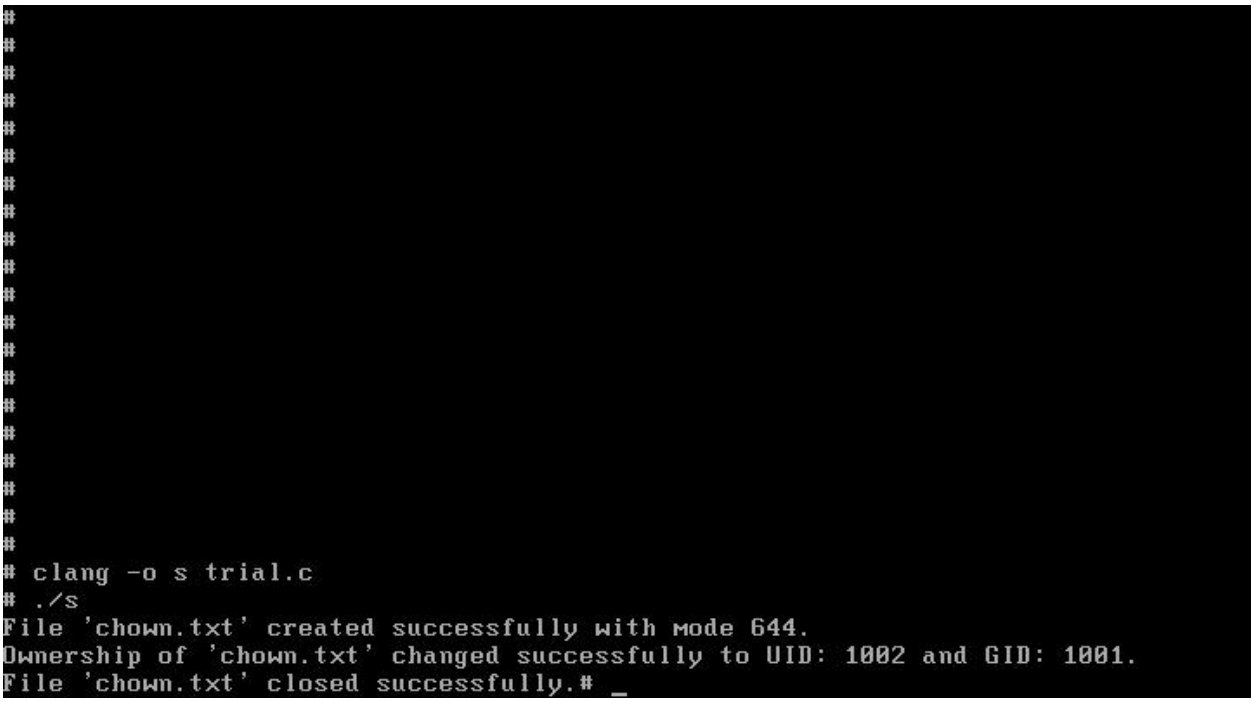

Image 8. The executable file executed with out errors

✧ Now the file's ownership has been changed and the file has been closed successfully.