Academic Report
Cover Page

CIS 657

OS Project 1

Ameya Kale
SU ID: 304967094
Email Address: akkale@syr.edu
October 6, 2022

Requirements

You will fix the first limitation in this project. We will discuss the second and third limitation in Project 2. Refer to TA's lecture if needed.

1. By default, Nachos can read only one user program name through the -x flag, no matter how many you specify with multiple -x flags. Modify main.cc so Nachos can take as many user programs via multiple -x flags.

Preliminary Requirement

Get a fresh copy of Nachos from nachos.tar. Refer to TA's lecture.
Download Project 1 files from Bb and place them in the right directories in your Nachos directory. These are main.cc and test1.tar.
Compile Nachos with the new main.cc. Unarchive test1.tar and compile all user programs in that directory.

Learning Objective

After completing this project, students can
1. modify the main program so it can read multiple user program names,

Procedure/Approach

1. In the main function of the main.cc file, a character pointer variable userProgName is initialized to NULL.
2. This variable is then used to store the program name arguments that are passed along with the command while running the program.
3. If we pass multiple program names, only the last name will be stored in this userProgName variable.
4. Therefore, we need to make some changes.
5. We need to redeclare the userProgName as an array.
6. But before we do that, we need to calculate the total number of -x flags that are being passed by the command while running the program.
    - To do that, we declare a counter variable called numOfxFlags.
    - We then compare the contents of the argv array with the string -x.
    - If we find a match, we then increament numOfxFlags by 1.

```
198     //Ameya: 1) We need to calculate the number of -x flags in the argument.
199     int numOfxFlags = 0;
200     for(int index = 0; index < argc; index++){
201         if (strcmp(argv[index], "-x") == 0) {
202             numOfxFlags += 1;
203         }
204     }
```

7. We then redeclare the userProgName as a character pointer array with the size of numOfxFlags, and initialize all the elements in the array to NULL.
8. We declare another global counter variable countOfxFlags to keep track of all -x flags in the userProgName array. We initialise this variable with 0.

```
206     //Ameya: 2) Then, we need to inistialse a pointer array userProgName with the number of the -x flags.
207     char *userProgName[numOfxFlags] = {NULL};
208
209     // Ameya: 3) Another global variable to keep track of total number of userProgName.
210     int countOfxFlags = 0;
```

9. Now we need to edit the code in the for loop that checks for all flags and performs the code respective to the flag. We are interested in editing the code that checks and executes for the -x flag.
10. Here, we update the value of userProgName at index = countOfxFlags, i.e. with the index that is available. We update this value with the next argument corresponding to -x (argv[i + 1]). We then increment the countOfxFlags by 1, indicating the next free index to store the next argument.

```
225     else if (strcmp(argv[i], "-x") == 0) {
226         ASSERT(i + 1 < argc);
227
228         // Ameya: 4) Update the userProgName with the available index with countOfFlags.
229         userProgName[countOfxFlags] = argv[i + 1];
230         countOfxFlags += 1;
231
232         i++;
233     }
```

11. To print the programs in the desired format, we simply loop over userProgName array and print the program names one by one.

```
333     // Ameya: 5) Print userProgName according to the given requirements.
334     for(int index = 0; index < numOfxFlags; index++){
335         if(userProgName[index]){
336             printf("Program [%d] = %s\n", index, userProgName[index]);
337         }
338         else{
339             break;
340         }
341     }
```

Conclusion

Now nachos can successfully read multiple program names through the -x flag.

Output

```
akkale@lcs-vc-cis486-2:~/nachos/code/build.linux$ ./nachos -x ../test1/add -x ../test1/halt
Program [0] = ../test1/add
Program [1] = ../test1/halt
```