

집중교육2- KERNEL API, LINKED LIST (10차과제 보고서)

201521032 한태희

1. 디자인과 구현

10차 과제에서는 커널 모드에서 링크드 리스트와 그 외 API를 활용하여 문자열을 데이터를 넣을 수 있는 링크드 리스트를 구현하고, addf, adde, delf, dela 명령이 가능하도록 구현하는 과제를 진행했다.

링크드 리스트는 리눅스에서 제공하는 API를 이용해 구현했다. list_head라는 구조체를 이용해 쌍방향 링크를 연결하고, list_head를 포함하는 노드를 정의해 해당 노드 안에 데이터를 같이 집어넣는 구조이다. 구현한 링크드 리스트는 기존에 예제로 주어졌던 형태와 한가지 차이가 있다. c언어는 문자열에 대해 자동적으로 메모리 할당, 해제가 되는 자료형이 없기 때문에 수동으로 char 배열에 동적 할당과 해제를 해주어야 한다. add가 될 때, 노드의 생성 후 문자열 공간을 할당하고, del이 될 때 노드의 해제 직전 문자열 공간도 해제하도록 구현했다.

유저에 대한 입력과 출력 통신은 직접 작성한 것이 아닌 주어진 코드에서 작성했다. 주어진 코드에선 유저의 입력이 발생하면 list_write 함수에서 local_buffer에 유저가 입력한 문자열이 저장되었다. 유저의 입력은 띄어쓰기로 구분된 2개의 단어에, 왼쪽의 단어는 명령어, 오른쪽의 단어는 리스트와 상호작용할 문자열이므로 위 2개의 정보를 strsep() 함수를 이용하여 추출했다. 명령어의 종류는 strcmp() 함수를 이용하여 정수형태로 저장한다. 그리고 4개의 명령어에 대하여 switch문을 이용해 4가지의 다른 함수를 실행하도록 했다.

addf 명령어일 경우, 새로운 노드를 생성한 후, 문자열을 strcpy() 함수를 이용해 복사한 다음 list_add() 함수를 이용해 노드를 리스트의 맨 앞에 집어넣는다. adde 명령어일 경우 addf와 동일하지만, list_add_tail() 함수를 이용해 노드를 리스트의 맨 뒤에 집어넣는다. delf 명령어일 경우, list_for_each_safe 함수를 이용하여 리스트를 시작부터 순차적으로 탐색하며, 매 노드마다 주어진 문자열과 같은 문자열을 가진 노드가 있는지 확인하고, 문자열이 일치한다면 해당 노드를 삭제하고 탐색을 종료한다. dela 명령어일 경우, 탐색을 종료하지 않고 문자열이 일치하는 모든 노드들을 삭제한다.

주어진 코드에서, 출력은 list_proc_show 함수에서 seq_puts() 함수를 이용해 seq_file에 문자열을 인자로 넘겨주는 구조이다. 따라서 list_for_each_entry 함수를 이용하여 모든 노드들을 순차적으로 탐색하며 순서대로 entry의 문자열들을 출력하도록 작성했다.

2. 검증

아래 스크린샷은 구현이 완료된 이후, 주어진 checker를 실행시킨 결과이다.

```

root@sce394vm:~/labs/lab10/api_assignment/checker# ./list-checker
[ 63.151923] list: loading out-of-tree module taints kernel.
module_exists.....passed [01/90]
folder_exists.....passed [01/90]
preview_exists.....passed [01/90]
management_exists.....passed [01/90]
preview_is_readable.....passed [01/90]
preview_is_not_writable.....failed [ 0/90]
management_is_writable.....passed [01/90]
management_is_not_readable.....passed [01/90]
addf_no_error.....passed [02/90]
addf_print.....passed [02/90]
addf_print_once.....passed [02/90]
adde_no_error.....passed [02/90]
adde_print.....passed [02/90]
adde_print_once.....passed [02/90]
addf_two_number.....passed [02/90]
addf_two_content.....passed [02/90]
addf_same_twice_number.....passed [02/90]
addf_same_twice_content.....passed [02/90]
addf_same_twice_content_number.....passed [02/90]
adde_two_number.....passed [02/90]
adde_two_content.....passed [02/90]
adde_same_twice_number.....passed [02/90]
adde_same_twice_content.....passed [02/90]
adde_same_twice_content_number.....passed [02/90]
addf_front_after_addf.....passed [02/90]
addf_front_after_adde.....passed [02/90]
[ 97.394760] systemd-journald[91]: Sent WATCHDOG=1 notification.
addf_front_after_adde_addf.....passed [02/90]
addf_front_after_addf_adde.....passed [02/90]
adde_end_after_addf.....passed [02/90]
addf_end_after_adde.....passed [02/90]
adde_end_after_adde_addf.....passed [02/90]
adde_end_after_addf_adde.....passed [02/90]
delf_no_error.....passed [02/90]
delf_removes_front.....passed [02/90]
delf_removes_end.....passed [02/90]
delf_front_number.....passed [02/90]
delf_front_content.....passed [02/90]
delf_first.....passed [02/90]
delf_none.....passed [02/90]
dela_one_number.....passed [02/90]
dela_one_content.....passed [02/90]
dela_one_number.....passed [02/90]
dela_one_content.....passed [02/90]
dela_all_five_number.....passed [02/90]
dela_all_five_content.....passed [02/90]
dela_none.....passed [02/90]
mix_number.....passed [03/90]
mix_content.....passed [03/90]

Total: [89/90]

```