PROJECT 1 REPORT

CSC 4320 – Operating Systems

Names: (report) Bronson Tharpe, (programmer 1) Yaorung Xiao, (programmer 2) Landon Wang, (programmer 3) Lin Lin, (demo) Monica Byrd

Email: [btharpe2@student.gsu.edu](mailto:btharpe2@student.gsu.edu)

**PART 1:**

**A screenshot of a computer

Description automatically generated**

**PART 2:**

Kernel log buffer contents after loading kernel module *simple*:

Graphical user interface, text

Description automatically generated

Kernel log buffer contents after loading module *simple*:

Text

Description automatically generated

**PART 3:**

Kernel log buffer contents after loading kernel module *simple-solution*:

A screenshot of a computer

Description automatically generated

Kernel log buffer contents after removing kernel module *simple-solution*:

A screenshot of a computer

Description automatically generated

**SOURCE CODE OF SIMPLE-SOLUTION.C:**

Top of Form

Bottom of Form

Top of Form

Bottom of Form

|  |  |
| --- | --- |
|  | #include <linux/init.h> |
|  | #include <linux/module.h> |
|  | #include <linux/kernel.h> |
|  | #include <linux/list.h> |
|  | #include <linux/slab.h> |
|  |  |
|  | struct birthday { |
|  | int month; |
|  | int day; |
|  | int year; |
|  | char \*name; |
|  | struct list\_head list; |
|  | }; |
|  |  |
|  |  |
|  | //The following defines and initializes a list\_head object named birthday\_list |
|  | static LIST\_HEAD(birthday\_list); |
|  |  |
|  | int simple\_init(void) { |
|  | struct birthday \*ptr; //for list traversal |
|  |  |
|  | // Create a linked list containing five struct birthday elements |
|  | // NOTE:THE NAME OF FIRST STRUCT BIRTHDAY SHOULD BE YOUR OWN NAME |
|  | struct birthday \*Landon; |
|  | struct birthday \*Lin; |
|  | struct birthday \*YaoRong; |
|  | struct birthday \*Brandson; |
|  | struct birthday \*Monica; |
|  |  |
|  | Landon = kmalloc(sizeof(\*Landon), GFP\_KERNEL); |
|  | Landon -> name = "Landon"; |
|  | Landon -> day = 14; |
|  | Landon -> month = 2; |
|  | Landon -> year = 2000; |
|  | INIT\_LIST\_HEAD(&Landon -> list); |
|  |  |
|  | Lin = kmalloc(sizeof(\*Lin), GFP\_KERNEL); |
|  | Lin -> name = "Lin "; |
|  | Lin -> day = 25; |
|  | Lin -> month = 8; |
|  | Lin -> year = 1999; |
|  | INIT\_LIST\_HEAD(&Lin -> list); |
|  |  |
|  | YaoRong = kmalloc(sizeof(\*YaoRong), GFP\_KERNEL); |
|  | YaoRong -> name = "YaoRong"; |
|  | YaoRong -> day = 8; |
|  | YaoRong -> month = 2; |
|  | YaoRong -> year = 1994; |
|  | INIT\_LIST\_HEAD(&YaoRong -> list); |
|  |  |
|  | Brandson = kmalloc(sizeof(\*Brandson), GFP\_KERNEL); |
|  | Brandson -> name = "Brandson"; |
|  | Brandson -> day = 7; |
|  | Brandson -> month = 23; |
|  | Brandson -> year = 2000; |
|  | INIT\_LIST\_HEAD(&Brandson -> list); |
|  |  |
|  | Monica = kmalloc(sizeof(\*Monica), GFP\_KERNEL); |
|  | Monica -> name = "Monica"; |
|  | Monica -> day = 8; |
|  | Monica -> month = 9; |
|  | Monica -> year = 1992; |
|  | INIT\_LIST\_HEAD(&Monica -> list); |
|  |  |
|  | list\_add\_tail(&Lin -> list, &birthday\_list); |
|  | list\_add\_tail(&Landon -> list, &birthday\_list); |
|  | list\_add\_tail(&YaoRong -> list, &birthday\_list); |
|  | list\_add\_tail(&Brandson -> list, &birthday\_list); |
|  | list\_add\_tail(&Monica -> list, &birthday\_list); |
|  |  |
|  | printk(KERN\_INFO "Loading Module\n"); |
|  |  |
|  | /\* Traverse the linked list \*/ |
|  | list\_for\_each\_entry(ptr, &birthday\_list, list) { |
|  | printk("Name: %s \t Birthday: %d %d %d", ptr -> name, ptr -> month, ptr -> day, ptr -> year); |
|  | } |
|  |  |
|  | printk(" "); |
|  |  |
|  | return 0; |
|  | } |
|  |  |
|  | void simple\_exit(void) { |
|  | /\* Remove the elements from the linked list and return the free memory back to the kernel \*/ |
|  | struct birthday \*ptr, \*next; |
|  |  |
|  | printk(KERN\_INFO "Cleaning and exiting module\n"); |
|  |  |
|  | list\_for\_each\_entry\_safe(ptr, next, &birthday\_list, list) { |
|  | //on each iteration ptr points to the next birthday struct |
|  | printk("Removing: %s", ptr -> name); |
|  | list\_del(&ptr -> list); |
|  | kfree(ptr); |
|  | } |
|  |  |
|  | printk(" "); |
|  | } |
|  |  |
|  | module\_init(simple\_init); |
|  | module\_exit(simple\_exit); |
|  |  |
|  | MODULE\_LICENSE("GPL"); |
|  | MODULE\_DESCRIPTION("Kernel Data Structures"); |
|  | MODULE\_AUTHOR("GSU\_CSC4320\_6320\_TH\_Fall2021"); |