

اضافه کردن یک فراخوان سیستمی به هسته‌ی لینوکس

هدف آزمایش: آشنایی با فراخوان‌های سیستمی در سیستم عامل لینوکس و نحوه‌ی اضافه کردن یک فراخوان سیستمی به هسته‌ی لینوکس

فراخوان‌های سیستمی برنامه‌ها و توابع از پیش تعریف شده درون سیستم عامل است که برنامه‌های کاربر از طریق فراخوانی آن‌ها قادر خواهند بود با هسته‌ی سیستم عامل ارتباط برقرار کرده و بر روی سخت‌افزار دستور اجرا کنند. تفاوت اساسی یک تابع که توسط کاربر نوشته و اجرا می‌شود با یک فراخوان سیستمی مُد اجرا شدن آن است. برنامه‌های کاربر در مُد کاربر اجرا می‌شوند که در این مُد دستورات ممتاز قابل اجرا نمی‌باشد در نتیجه برنامه‌های کاربر قادر نخواهند بود که به سخت‌افزارهای سیستم به طور مستقیم دسترسی داشته باشد، در حالی که فراخوان‌های سیستمی در مُد سیستم عامل اجرا می‌شوند و به همین دلیل بدون محدودیت به تمام سخت‌افزارهای سیستم به صورت مستقیم دسترسی داشته و سرعت اجرای بالاتری دارند.

مراحل کار: برای اضافه کردن یک فراخوان سیستمی به هسته سیستم عامل لینوکس باید ابتدا مراحل زیر را به ترتیب انجام دهید:

(۱) به‌روزرسانی بسته‌های مورد نیاز برای کامپایل کردن هسته‌ی لینوکس که شامل موارد زیر می‌باشد:

kernel-package, libncurses5-dev, wget, bzip2

برای به‌روزرسانی این بسته‌ها از دستورهای زیر استفاده کنید:

```
sudo apt-get update; sudo apt-get upgrade
```

```
sudo apt-get install [package name]
```

(۲) دانلود آخرین نسخه‌ی هسته‌ی لینوکس از سایت www.kernel.org. فایل مربوطه را از حالت فشرده خارج کرده (دستور tar kjf) و در شاخه‌ی /usr/src کپی کنید (دستور CP) و وارد آن شاخه شوید (دستور cd). بهتر است خط فرمان خود را root کنید، زیرا اکثر دستورات این آزمایش نیاز به دسترسی root دارد.

(۳) یک نام برای فراخوان سیستمی خود در نظر بگیرید مثلاً `mysyscall`. سپس یک شاخه جدید با همین نام ساخته و در آن شاخه یک فایل C و با یک نام `syscall.c` ایجاد کنید. در فایل جدید یک تابع با نام

فراخوان سیستمی ایجاد کنید و در آن یک مقدار صحیح دریافت کرده و ده برابر آن را برگردانید. توجه داشته باشید که ابتدای نام فایل باید “sys_” اضافه شود و قبل مقدار برگشتی کلمه کلیدی “asm linkage” وجود داشته باشد. برای مقدار برگشتی تابع نیز از **long** استفاده کنید مثلاً `asm linkage long sys_mysyscall(int test)`. در فایلی که به زبان C به عنوان فراخوان سیستمی می‌نویسید باید سرفایل‌های زیر را استفاده کنید تا در زمان کامپایل کردن هسته‌ی جدید فراخوان سیستمی به درستی کامپایل شود.

<linux/kernel.h>, <linux/linkage.h>, <linux/syscalls.h>

(۴) به انتهای فایلی با نام `syscall_32.tbl` که در شاخه‌ی `/arch/x86/syscalls/` قرار دارد مشخصات فراخوان سیستمی جدید را اضافه کنید. مشخصاتی که باید برای یک فراخوان سیستمی وارد شود شامل شماره‌ی فراخوان سیستمی، نام و ... می‌باشد.

(۵) به انتهای فایلی با نام `syscalls.h` که در شاخه‌ی `/include/linux/` قرار دارد امضاء تابع فراخوان سیستمی را اضافه کنید.

(۶) در شاخه‌ی جدیدی که برای فراخوان سیستمی جدید در نظر گرفته‌اید یک فایل جدید به نام `Makefile` ایجاد کنید. این فایل هنگام کامپایل هسته‌ی جدید فراخوان سیستمی جدید را به صورت یک `object` به هسته اضافه می‌کند. پس از ایجاد کردن فایل مربوطه محتویات آن را برای کامپایل کردن فراخوان سیستمی جدید آماده کنید. (نیاز است که یک دستور درون فایل اضافه کنید)

(۷) فایلی به نام `Makefile` که در شاخه‌ی اصلی قرار دارد طوری تغییر دهید که به هنگام کامپایل کردن هسته‌ی جدید فراخوان سیستمی جدید را نیز کامپایل کند. (باید به قسمت خاصی از این فایل نام فراخوان سیستمی جدید را اضافه کنید)

(۸) **پیکربندی هسته:** با استفاده از دستور `sudo make menuconfig` وارد منوی پیکربندی هسته‌ی جدید شوید و تنظیمات لازم از نظر پیکربندی را برای هسته‌ی جدید انجام دهید (متناسب با سخت‌افزار و خواست خودتان) و پیکربندی جدید را در یک فایل با نام جدید ذخیره کنید.

(۹) سپس فایل جدید را به فایل پیکربندی اصلی منتقل کنید (دستور `mv.....`).

(۱۰) **کامپایل و نصب هسته‌ی جدید:** برای کامپایل کردن هسته‌ی جدید باید از دستور `sudo make` استفاده کنید که این قسمت معمولاً زمان‌بر می‌باشد. برای این که زمان کمتر را صرف کامپایل کردن

کنید می‌توانید از گزینه‌ی j- همراه با دستور `sudo make` استفاده کنید (این گزینه مخصوص سیستم‌های با پردازنده‌های چند هسته‌ای می‌باشد).

سپس با استفاده از دستورهای زیر هسته‌ی کامپایل شده را به صورت ماژولار بر روی سیستم نصب کنید.

```
sudo make modules_install
sudo make install
```

پس از اجرای دستورات زیر باید در شاخه `/boot/` فایل‌های زیر به صورت خودکار به وجود آمده باشند:

- `System.map-<kernel-version>`
- `vmlinuz-<kernel-version>`
- `initrd.img-<kernel-version>`
- `config-<kernel-version>`

در صورتی که فایل `initrd.img-<kernel-version>` در شاخه `/boot/` ایجاد نشده باشد هسته کامپایل شده نصب نشده است در این حالت با استفاده از دستور زیر می‌توانید فایل را ایجاد کرده و هسته را نصب کنید.

```
sudo update-initramfs -u -k <kernel-version>
```

(۱۱) در نهایت با استفاده از دستور زیر می‌توانید grub لینوکس را به‌روزرسانی کنید که این کار باعث می‌شود که هسته‌ی جدید به منوی اصلی سیستم اضافه شده و شما بتوانید سیستم عامل لینوکس خود را با استفاده از هسته جدید راه‌اندازی کنید.

```
sudo update-grub
```

(۱۲) پس از انجام موفقیت‌آمیز مراحل فوق لینوکس خود را راه‌اندازی مجدد کرده و با هسته‌ی جدید که روی سیستم نصب کرده‌اید لینوکس را راه‌اندازی کنید. سپس با استفاده از اجرای دستور `uname -r` در محیط ترمینال از نصب نسخه‌ی جدید هسته‌ی لینوکس بر روی سیستم خود اطمینان حاصل نمایید.

(۱۳) برنامه‌ای به زبان C بنویسید و فراخوان سیستمی جدیدی که به هسته‌ی لینوکس اضافه کرده‌اید را امتحان کنید. برای این که بتوانید این فراخوان سیستمی را فراخوانی کنید باید از تابع `syscall(...)` استفاده کنید. توجه داشته باشید که در این برنامه باید سرفایل `<unistd.h>` را نیز استفاده کنید.

راهنمایی: تمامی فایل‌هایی که در این آزمایش باید تغییر دهید برای تغییرشان نیاز دارید که از دسترسی root استفاده کنید. در صورتی که از دستور vim استفاده کنید می‌توانید در خط فرمان root فایل‌های گفته شده را اصلاح و ذخیره کنید. توجه داشته باشید که برای استفاده از دستور vim باید پکیج مربوط به آن را نصب کنید که برای نصب آن نیز باید دستور زیر را در محیط ترمینال اجرا کنید که پکیج مربوطه به صورت آنلاین بر روی سیستم شما نصب شود.

موارد خواسته شده در گزارش کار: در گزارش کار خود در مورد انواع فراخوان‌های سیستمی سیستم عامل لینوکس توضیح دهید. مراحل کار را به صورت کامل در گزارش کار آورده و در مورد بخش‌ها و توابعی که در دستور کار توضیح داده نشده حتماً به صورت کامل توضیح دهید و از مراحل کار خود تصاویری در گزارش کار بیاورید.

موفق باشید

پارسه