باسمه تعالی

آزمایشگاه سیستم‌عامل

پروژه‌ی دوم: فراخوان سیستمی

تاریخ تحویل : ۱۸ فروردین

اهداف پروژه

- آشنایی با پیاده‌سازی فراخوان‌های سیستمی در هسته‌ی لینوکس
- آشنایی با لیست پیوندی هسته‌ی لینوکس
- آشنایی با ساختار داده‌ی task_struct و نحوه‌ی ذخیره‌سازی پردازه‌ها در هسته‌ی لینوکس
- آشنایی با ساختار داده‌ی files_struct
- آشنایی با فراخوانی سیستمی open

شرح آزمایش

در این آزمایش تعدادی فراخوانی سیستمی به هسته اضافه خواهد شد. در این فراخوانی‌ها پردازش‌هایی بر پردازه‌های موجود در هسته و داده‌هایشان انجام می‌شوند که از سطح کاربر قابل انجام نیستند. برای پردازش، ابتدا باید داده‌ها در یک لیست پیوندی کپی شوند و سپس عملیات روی این لیست انجام شود. در اینجا منظور از لیست پیوندی، لیست پیوندی هسته‌ی لینوکس است که با آن آشنا خواهید شد. تمامی مراحل کار باید در گزارش کار همراه با فایل‌هایی که آپلود می‌کنید موجود باشند.

نحوه‌ی ذخیره‌ی اطلاعات پردازه‌ها در هسته

در ابتدایی که یک سیستم شروع به کار می‌کند تنها یک پردازه در هسته‌ی لینوکس وجود دارد که init_task نام دارد. همه‌ی پردازه‌های موجود در هسته‌ی لینوکس از این پردازه‌ی اولیه یا فرزندان آن ایجاد می‌شوند. ساختار پردازه‌ها در لینوکس به این صورت است که هر پردازه یک پدر و تعدادی فرزند دارد که در یک لیست در داخل ساختار داده‌ی task_struct نگهداری

می‌شوند. در این ساختار داده برای پردازه‌ها مشخصه‌های دیگری نیز وجود دارد. در این آزمایش شما با فایل‌ها[1] از نوع ساختار داده‌ی files_struct و شناسه‌ی پردازه‌ها[2] کار خواهید داشت.

نحوه‌ی اضافه کردن یک فراخوان سیستمی

برای انجام این کار لینک و مستندات زیادی در اینترنت و منابع دیگر موجود است. شما باید به جست‌وجوی روش درست انجام این کار بپردازید و پس از آزمودن روش، مستند مناسبی از نحوه‌ی اضافه کردن فراخوان سیستمی را در گزارش خود بیاورید. گزارش شما باید شامل تمامی مراحل اضافه کردن فراخوانی سیستمی و همینطور مراحل بعد باشد.

آشنایی با فراخوانی سیستمی open

در هسته‌ی لینوکس فایل‌ها با شناسه‌ی فایل[3] مربوطه‌شان شناخته می‌شوند و پردازه‌ها با این شناسه‌ها به فایل‌ها دسترسی می‌یابند. فراخوانی سیستمی sys_open در هسته‌ی لینوکس یک فایل را با آدرس مشخص‌شده باز می‌کند و یا در صورت عدم وجود، آن را ایجاد می‌کند. مقدار بازگشتی این فراخوانی سیستمی پس از فراخوانی توسط یک پردازه، یک شناسه‌ی فایل می‌باشد که به فایل باز شده اشاره دارد. اطلاعات بیشتر در رابطه با این فراخوانی سیستمی در اینترنت موجود است. هر پردازه یک مشخصه برای نگه داشتن تمام فایل‌های باز شده‌اش دارد که این مشخصه حاوی یک آرایه از تمام شناسه‌ی فایل‌های باز شده است. با پردازش روی فایل‌های یک پردازه، به لیستی از شناسه‌های فایل‌هایش می‌رسیم.

پیاده‌سازی فراخوانی‌های سیستمی

در این قسمت قصد داریم تا با استفاده از چند فراخوانی سیستمی و یک متغیر سراسری از نوع لیست پیوندی هسته‌ی لینوکس، فایل‌های بازشده توسط پردازه‌های مختلف را بر حسب شناسه‌ی فایلشان مرتب کنیم.

فراخوانی‌های سیستمی که باید پیاده‌سازی کنید از این قرار می‌باشند:

---

[1] Files
[2] pid
[3] file descriptor

- asmlinkage long sys_init_process_list(pid_t p)

این تابع با گرفتن شناسه‌ی یک پردازه، پردازه را در یک لیست پیوندی لینوکس ذخیره می‌کند و برای هر پردازه در این لیست پیوندی، یک لیست پیوندی دیگر برای شناسه‌های فایل‌هایش ایجاد می‌کند. تابع این کار را برای فرزندان پردازه نیز تکرار می‌کند. این ذخیره‌سازی به صورت کپی کردن انجام می‌شود تا در صورت تغییر پردازه‌ها در آینده، لیست دچار تغییر نشود.

در صورتی که پردازه‌ای با این شناسه پیدا نشود، باید لیست با زیردرخت پردازه‌ی init_task مقدار دهی شود. در این حالت خروجی تابع ۱ خواهد بود. در صورت عملیات موفق خروجی تابع ۰ می‌باشد.

- asmlinkage long sys_sort_process_list(void)

این تابع محتویات لیست پیوندی را بر اساس شناسه‌های پردازه‌ها به صورت صعودی مرتب می‌کند و مقدار ۰ به عنوان خروجی بازگردانده می شود. در صورتی که لیست از قبل مقدار دهی نشده باشد، مطابق چیزی که در تابع قبل گفته شد عمل شود.

دقت داشته باشید که این مرتب‌سازی باید به وسیله‌ی جابجا کردن خانه‌های لیست پیوندی انجام شود و جابجا کردن فیلدهای داده مد نظر نیست.

- asmlinkage long sys_print_process_list(void)

در این تابع عناصر لیست به صورت جفت‌های شناسه‌ی پردازه و طول لیست پیوندی شناسه‌های فایل (pid, len of file_list)، با استفاده از دستور printk نمایش داده می‌شوند. در مواجه با لیست مقداردهی نشده، مشابه تابع‌های بالا عمل شود. نتیجه‌ی این تابع باید به صورت مرتب شده نمایش داده شود.

- asmlinkage long sys_clear_process_list(void)

این تابع لیست پیوندی را پاک می‌کند.

مرتب‌سازی لیست شناسه‌های فایل هر پردازه (امتیازی)

در این قسمت، فراخوانی سیستمی‌ای را پیاده سازی می‌کنید که لیست شناسه‌های فایل هر پردازه را به صورت نزولی مرتب می‌کند. ساختار این فراخوانی به شکل زیر می‌باشد:

- asmlinkage long sys_sort_file_descriptor_list (void)

پس از اجرای این تابع، اگر فراخوانی چاپِ لیست صدا زده شود، خروجی به صورت جفت‌های شناسه‌ی پردازه و بزرگ‌ترین شناسه‌ی فایل آن پردازه (pid, maximum file descriptor)، با استفاده از دستور printk نمایش داده می‌شوند.

نکاتی در رابطه با فراخوانی‌های سیستمی

- برای توابع خود یک برنامه بنویسید تا از صحت کارکرد آن‌ها اطمینان حاصل کنید.
- منظور از لیست پیوندی هسته‌ی لینوکس struct_list_head می باشد.
- برای ردیابی روال فراخوانی‌ها، پیغام‌های مناسبی در جاهای مناسب چاپ کنید.

سایر نکات

- فقط بخش‌های تغییریافته‌ی هسته را به همراه گزارش خود آپلود کنید.
- تمام مراحل کار را در گزارش کار خود بیاورید.

موفق باشید