



نیمسال دوم ۱۴۰۰-۱۴۰۱

مدرس: دکتر مجتبی رفیعی

اصول سیستم‌های عامل

جلسه ۱۶

نگارنده: نگار حسینعلی پور

۱ اردیبهشت ۱۴۰۱

فهرست مطالب

- | | |
|---|------------------------------------|
| ۱ | ۱ ایجاد فرآیند |
| ۳ | ۲ تخصیص منابع فرآیند فرزند |
| ۳ | ۳ نحوه اجرای فرآیندهای پدر و فرزند |
| ۳ | ۴ فضای آدرس فرآیند فرزند |
| ۴ | ۵ خاتمه فرآیند |

۱ ایجاد فرآیند

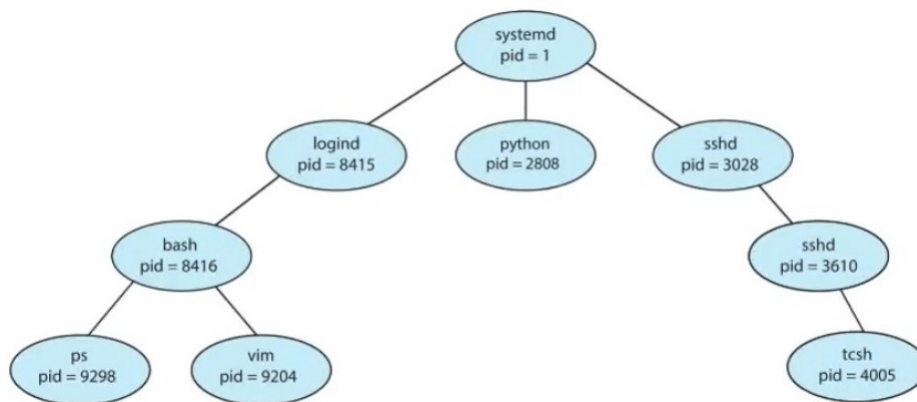
یک فرآیند در طول اجرای خود ممکن است چندین فرآیند جدید تولید کند. به فرآیند اصلی، فرآیند پدر^۱ و به فرآیندهای جدید تولید شده توسط آن، فرآیندهای فرزند^۲ اطلاق می‌شود.

¹Parent

²Children

بنابراین در یک سیستم می‌توان برای فرایندها، یک ساختار درختی در نظر گرفت. هر فرآیند در سیستم یک شناسه یکتا^۳ دارد که به‌طور مختصر به آن PID اطلاق می‌شود. این شناسه یک عدد صحیح است و به عنوان یک شاخص^۴ برای دسترسی به فرآیند و آگاهی از ویژگی‌های آن توسط سیستم عامل استفاده می‌شود.

شکل زیر یک درخت فرآیند نمونه در سیستم عامل لینوکس را نشان می‌دهد:



فرآیند systemd که همیشه دارای شناسه فرآیند ۱ (pid=1) است، ریشه برای تمامی فرایندهای کاربر است. این فرآیند مسئولیت آماده‌سازی فضای کاربر و مدیریت همه‌ی فرایندهای بعد از خود را دارد. در واقع، هنگامی که سیستم بوت می‌شود، فرآیند systemd شروع به ایجاد فرایندهایی برای ارائه سرویس‌هایی نظیر وب، ssh، چاپگر و... می‌کند.

اطلاعات جانبی

حرف d در انتهای systemd از کلمه daemon گرفته شده و به‌طور معمول به برنامه‌هایی اطلاق می‌شود که به‌طور پیوسته و در پس‌زمینه سیستم در حال اجرا بوده و به درخواست‌های مربوط به سرویس‌ها پاسخ می‌دهد.

در شکل فوق، دو فرآیند logind و sshd به عنوان فرایندهای فرزند توسط فرآیند systemd به شرح زیر ایجاد شده است:

- **فرآیند logind:** این فرآیند مدیریت کاربرانی را بر عهده دارد که به طور مستقیم لاگین کرده‌اند. در شکل فوق یک کاربر به سیستم لاگین کرده و از bash shell استفاده می‌کند. به طور دقیق‌تر، کاربر با استفاده از رابط command-line دو فرآیند PS (گزارش‌گیری از فرایندهای فعال در سیستم) و فرآیند VIM (ویرایشگر متن) ایجاد کرده است.

- **فرآیند sshd:** این فرآیند مدیریت کاربرانی را بر عهده دارد که با استفاده از ssh^۵ به سیستم متصل شده‌اند.

اطلاعات جانبی

با استفاده از دستور ps (Process State) به صورت ps -el می‌توان لیست فرایندهای فعال در سیستم و ساختار درختی مربوط به آن را به کمک ستون PID و PPID مشاهده و بازترسیم کرد.

^۳Process Identifier

^۴Index

^۵Secure Shell

۲ تخصیص منابع فرزند

فرآیند فرزند برای در اختیار گرفتن منابع سیستم دو امکان دارد:

۱. به طور مستقیم و بدون محدودیت از سیستم عامل منابع مورد نیاز خود را درخواست کند،
۲. به زیر مجموعه‌ای از فرآیند پدر خود محدود باشد. در چنین حالتی فرآیند پدر ممکن است:

- منابع را بین فرآیندهای فرزند خود افزایش دهد،
- تمامی منابع را بین فرزندان به اشتراک بگذارد.

لازم به ذکر است که فرآیند پدر علاوه بر تأمین منابع فیزیکی و منطقی برای فرآیندهای فرزند، ارسال ورودی یا مقداره‌های اولیه مورد نیاز به آنها را نیز برعهده دارد.

مثال: گرفتن ورودی فرآیند فرزند از فرآیند پدر:

فرآیندی را در نظر بگیرید که عملکرد آن نمایش محتوای hw1.c روی صفحه ترمینال است. هنگامی که چنین فرآیندی ایجاد می‌شود، نام فایل hw1.c را از فرآیند پدر خود دریافت و سپس فایل را باز کرده و محتوای آن را در خروجی می‌نویسد. حتی ممکن است فرآیند فرزند نام دستگاه خروجی را نیز به عنوان ورودی دریافت کند. در برخی سیستم‌عامل‌ها، فرآیند پدر ممکن است منابع دو فایل باز hw1.c و دستگاه ترمینال را در اختیار فرآیند فرزند قرار داده و او به راحتی وظیفه خود را تکمیل کند.

۳ نحوه اجرای فرآیندهای پدر و فرزند

هنگامی که یک فرآیند جدید ایجاد می‌شود، دو امکان برای اجرای آن فرآیند و فرآیند پدر وجود دارد:

- فرآیند پدر همروند با اجرای فرآیند فرزند است،
- فرآیند پدر تا خاتمه چند یا همه فرزندان منتظر می‌ماند.

۴ فضای آدرس فرزند

برای تخصیص فضای آدرس به فرآیند جدید (فرزند) دو امکان وجود دارد:

- فضای آدرس فرآیند فرزند یک نسخه تکراری از فرآیند پدر است؛ بدین معنا که برنامه و داده یکسان با فرآیند پدر خود دارد،
- در فرآیند فرزند یک برنامه جدید بارگزاری می‌شود.

فرآیند فرزند متناسب با استراتژی اتخاذی برای ایجاد آن می‌تواند اولویت‌ها، خصیصه‌های زمانبندی، منابع مشخص شده برای فرآیند پدر (مثل فایل‌های باز) و... را به ارث ببرد.

جمع بندی: تعامل فرآیندهای فرزند و پدر را می‌توان از سه منظر زیر رده‌بندی کرد:

- ۱- تخصیص منابع: مستقل، اشتراکی، افزایش بین فرزندان،
- ۲- اجرای فرآیند: همروند با پدر، اجرای غیر همروند،
- ۳- فضای آدرس (داده و برنامه): مستقل، نسخه تکراری.

۵ خاتمه فرآیند

یک فرآیند در چرخه اجرایی خود می‌تواند به دلایل مختلفی اعم از موارد زیر خاتمه یابد:

- **خاتمه عادی:** در این حالت، یک فرآیند بعد از اجرای آخرین دستورالعمل خودش، فراخوان سیستمی `exit()` را فراخوانی می‌کند. در واقع با این فراخوانی، فرآیند به سیستم عامل فرمان می‌دهد که او را از مجموعه فرآیندهای فعال حذف کند. این فراخوانی توالی را با خود به همراه دارد که در آن وضعیت فرآیند به فرآیند پدرش اطلاع رسانی می‌شود، همچنین کلیه منابع تخصیص یافته نظیر حافظه فیزیکی/مجازی، فایل‌های باز، بافرهای I/O و... نیز به منابع آزاد اضافه می‌شود.

- **خاتمه فرآیند توسط فرآیند پدر:** یک فرآیند ممکن است با استفاده از فراخوان‌های سیستمی توسط فرآیند دیگری خاتمه یابد. به‌طور معمول چنین فراخوان‌هایی توسط فرآیند پدر (parent) انجام می‌شود.

- **خاتمه توسط کاربر یا برنامه کاربردی خرابکار:** به چنین خاتمه‌ای اصطلاحاً `kill` کردن یک فرآیند اطلاق می‌شود.

ضرورت آگاهی شناسه فرآیند فرزند توسط فرآیند پدر: یک فرآیند اگر بخواهد فرزند خود را خاتمه دهد باید شناسه آن را بداند. بنابراین وقتی یک فرآیند جدید به عنوان مثال با فراخوان سیستمی نظیر `fork()` ایجاد می‌شود، شناسه آن فرآیند برای فرآیند پدرش ارسال می‌شود.

اطلاعات تکمیلی

خروجی حاصل از فراخوانی `fork` توسط یک فرآیند، یک شناسه فرآیند (PID) است که می‌تواند مقادیر زیر را به خود بگیرد:

۱- $PID > 0$: فرآیند جدید به درستی ایجاد شده از این به بعد فرآیندی با شناسه PID معتبر است،

۲- $PID < 0$: شناسه نامعتبر بوده و فرآیند جدید با موفقیت ایجاد نشده است،

۳- $PID = 0$: اگر سورس فرآیند پدر و فرزند یکی باشد، مقدار خروجی دستور `fork` برای فرآیند فرزند حاوی مقدار صفر است. لازم به ذکر است که برای فرآیند پدر، خروجی تابع `fork`، PID فرآیند جدید است.