

1. برای اجرای فراخوان‌های سیستم جدید clone و join در xv6 ، ابتدا فایل syscall.h را با اضافه کردن آن‌ها و اختصاص شماره‌های (۲۳ و ۲۴) به‌روزرسانی می‌کنیم.
2. در فایل syscall.c ، این فراخوان‌های سیستم را تعریف می‌کنیم.
3. کاربران می‌توانند از طریق sysproc.c به آن‌ها دسترسی پیدا کنند و ما آن‌ها را در user.h با استفاده از SYSCALL(clone) و SYSCALL(join) اعلام می‌کنیم.
4. ما همچنین توابع مربوطه مانند thread\_create ، thread\_join ، lock\_init ، lock\_acquire و lock\_release را در user.h تعریف می‌کنیم. این توابع که برای استفاده از مفهوم قفل‌ها حیاتی هستند، با استفاده از ساختار lock\_t در user.h تنظیم می‌شوند.
5. پیاده‌سازی در sysproc.c شامل ارسال ورودی‌ها از طریق argint می‌باشد.
6. در فایل proc.c ، ما توابع clone و join را پیاده‌سازی می‌کنیم که در آن، clone یک فرایند جدید با اطلاعات انباره مشخص شده ایجاد می‌کند و join به دنبال فرایندهای فرزندی در جدول فرایندها می‌گردد و آن‌ها را به حالت‌های زامبی برای پاک شدن تبدیل می‌کند.
7. ما این توابع را در proc.h و defs.h تعریف می‌کنیم و یک خاصیت جدید threadstack را در یک ساختار برای نگهداری آدرس‌های نخ اضافه می‌کنیم.
8. توابع جدید برای کتابخانه در ulib.c توضیح داده شده‌اند.
9. برای آزمایش، ما یک فایل جدید با نام test\_threads.c ایجاد می‌کنیم. در اینجا، کاربران می‌توانند thread\_create را فرا بخوانند، که توابعی را اجرا می‌کند که در ابتدا یک شمارنده را که صفر تنظیم شده است، افزایش می‌دهند.
10. از آنجایی که ما از قفل‌هایی که قبلاً تعریف کرده‌ایم استفاده می‌کنیم، ما امنیت نخ را تضمین می‌کنیم و انتظار داریم که شمارنده در پایان به ۳ برسد.

<https://github.com/Parsavazifeh/Operating-System.git>

SeaBIOS (version 1.15.0-1)

iPXE (https://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+1FF8B4A0+1FECB4A0 CA00

Booting from Hard Disk..xv6...

cpu0: starting 0

sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58

init: starting sh

\$ test\_threads

Thread 2 increased counter to 1

Thread 3 increased counter to 2

Thread 1 increased counter to 3

Final Counter Value: 3

\$ █