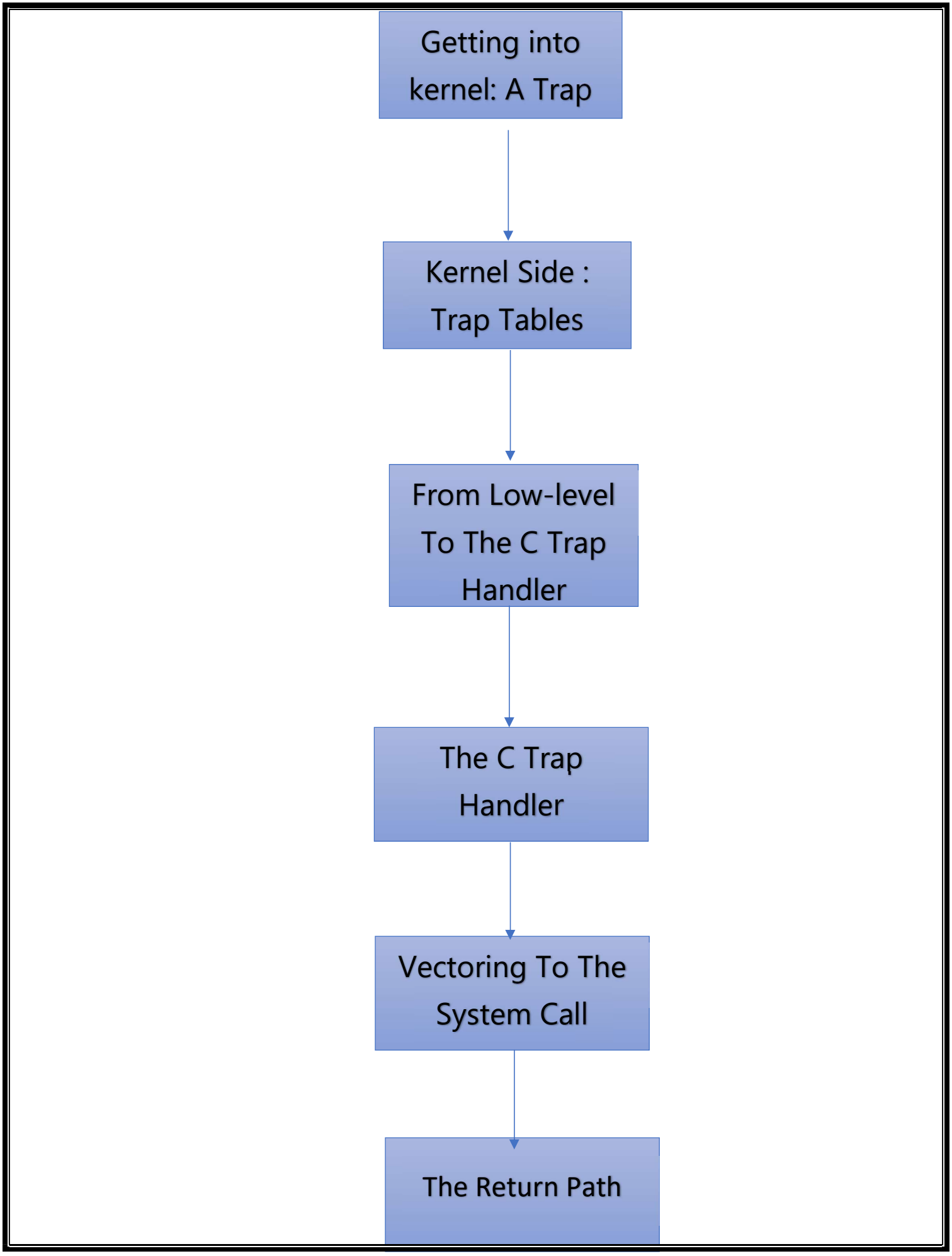
مراحل اجرایی یک System Call به این صورت مرحله بندی میشود:

- system call: System Call Overview به سیستم عامل اجازه میدهد که از طرف کاربر کد را به شیوه ای امن به درخواست دادن به کرنل اجرا کند.

- Getting Into Kernel: A Trap : در اولین مرحله از سطح کاربر یک برنامه شروع میشود. (برای مثال در یک برنامه از تابع ()Write برای خواندن از فایل فراخوانی میشود) کاری که کتابخانه انجام میدهد این است که آرگومان های مناسب را در Register مربوطه قرار میدهد و نوعی از دستورات Trap را ارسال میکند. که این کد در فایل Usys.S قرار دارد. این فایل شامل ($ int) است که به سخت افزار نشان میدهد که نوع Trap چیست.
- Kernel Side: Trap Tables: هنگامی که int در حال اجرا است سخت افزار از طرف caller کار های زیادی انجام میدهد.یکی از کارهای مهمی که سخت افزار این است که سطح دسترسی CPU را در حالت کرنل افزایش می دهد. از حالت CPL3 به حالت CPL0 تغییر وضعیت میدهد. کارمهم دومی که در سخت افزار انجام میشود این است که کنترل Trap Vector را از سیستم انتقال می دهد تا سخت افزار را قادر بسازد تا چه کدی را برای هر Trap اجرا که در فایل Trap.c انجام میشود.

- From Low-level To The C Trap Handler : پس از این که سیستم عامل Trap Handler را راه اندازی کرد سیستم عامل آماده میشود که توسط int این بخش را درست کند.ابتدا سخت افزار تعداد task را اعلام میکند. ابتدا تسک های که برای نرم افزار سخت و یا غیر ممکن هستند را انجام می دهد مانند (Saving the current pc , stack pointer) که توضیحات در فایل X86.h مشهود است. همانطور که پیداست در ساختار Trap Frame بخشی از آن ها توسط سخت افزار و بقیه آن ها توسط سیستم عامل پر میشود.
- The C Trap Handler : هنگامی که جزییات سطح پایین تنظیمات Trap Frame تمام شد، کد اسمبلی در یک Generic C Trap Handler که ()Trap نامیده می شود که در یک اشاره گر به Trap Frame ارسال میکند.این Trap Handler برای هر نوع interrupt, Trap فراخوانی میشود.
- Vectoring To The System Call: بلاخره به سراغ Syscall که در فایل syscall.c است میرویم. شماره System call به رجیستر eax% ارسال میشود و می توانیم این شماره را از Trap Frame بدست آوریم تا روند مناسب آن در که در system call فراخوانی می کنیم. پس از آنکه شماره System call چک شد pointed-to برای مدیریت فراخوانی استفاده میشود.
- Return Path : ابتدا System call یک عدد بر میگرداند که کدی که در Syscall است که آن را برمیدارد در eax% قرار میدهد. سپس کد به ()Trap باز می گردد که به سادگی به جایی که از داخل فراخوانی شده بر می گردد.
- دیاگرام مراحل اجرایی یک System Call به شرح زیر است:


Getting into kernel: A Trap
Kernel Side : Trap Tables
From Low-level To The C Trap Handler
The C Trap Handler
Vectoring To The System Call
The Return Path