۳)‌ آیا باقی تله‌ها را نمی‌توان با سطح دسترسی DPL\_USER فعال نمود؟ چرا؟

نه نمی‌توان چنین کاری را انجام داد زیرا برنامه‌ها سمت کاربر غیر قابل اعتماد هستند و اگر به آن‌ها اجازه دهیم که از دیگر trapها استفاده کنند ممکن است در کار سیستم‌عامل اختلال ایجاد کنند. برای مثال یک برنامه می‌تواند به اشتباه ارور تقسیم بر صفر را گزارش کند.

۴) در صورت تغییر سطح دسترسی ss و esp بر روی استک push می‌شود و در غیر این صورت نمی‌‌شود. چرا؟

رجیستر esp برای نگه داشتن سر استک و ss برای نگه داشتن بلاکی از مموری که استک در آن ذخیره شده استفاده می‌شود. هنگامی که سطح دسترسی در فراخوانی trap تغییر می‌کند این مقادیر در ابتدای ‌‌استک کرنل push می‌شوند تا بتوان پس از پایان trap به برنامه اصلی بازگشت.

نحوه‌ی اضافه کردن system call جدید: (با نام name)

۱- ابتدا کد مربوط به system call جدید را با یک macro به نام SYS\_name در فایل syscall.h اضافه می‌کنیم.

۲- سپس پیاده سازی مربوط به system call را در sysproc.c یا sysfile.c با نام sys\_name انجام می‌دهیم.

۳- سپس کد system call را به پیاده‌سازی آن در syscall.c مپ می کنیم.

۴- عبارت (SYSCALL(name را به فایل usys.S اضافه می‌کنیم.

۵- در نهایت prototype مربوط به آن را به user.h اضافه می‌کنیم تا توسط برنامه‌های سمت کاربر قابل استفاده باشد.

فراخوانی سیستمی calculate\_sum\_of\_digits: برای گرفتن ورودی مربوط به این system call از trapframe مربوط به process کنونی استفاده می‌کنیم.

myproc() -> tf -> ebx

فراخوانی get\_parent\_id: توضیح خاصی ندارد

myproc() -> parent -> pid