**گزارش فاز اول پروژه سیستم‌های عامل**

**اشکان شکیبا (۹۹۳۱۰۳۰)**

**سوال اول**

ابتدا تابع kinit حافظه را آزاد می‌کند. سپس با اجرای تابع kvminit یک پیج تیبل ساخته می‌شود تا کرنل بتواند از فضای آدرس‌های مختلف استفاده کند. در ادامه تابع kvminithart برای هر هسته پردازنده اجرا می‌شود و پیج تیبل را راه‌اندازی و در نهایت فلاش می‌کند. پس از آن تابع procinit فراخوانی شده و جدول پردازه‌ها را راه‌اندازی می‌کند. سپس تابع trapinit یک لاک برای ساخت trapها می‌سازد. در ادامه تابع trapinithart اجرا می‌شود که وظیفه مدیریت فراخوانی‌های سیستمی و وقفه‌ها را دارد. پس از آن تابع plicinit کنترل‌گر وقفه‌ها را راه‌اندازی می‌کند. سپس تابع plicinithart اجرا می‌شود که مربوط به وقفه‌های دستگاه‌هاست. در ادامه تابع binit اجرا شده و یک بافر کش مرتب‌شده می‌سازد. پس از آن تابع iinit با لاک کردن inodeها دسترسی به آنها را مدیریت می‌کند. سپس تابع fileinit با لاک کردن ftable فایل‌ها را نگهداری می‌کند. در ادامه با اجرا شدن تابع virtio\_disk\_init درایورها ساخته می‌شوند.

پس از همه این مراحل، نوبت به اجرای تابع userinit می‌رسد. با اجرای این تابع ابتدا متغیر p که اشاره‌گری از نوع پردازه است ساخته شده و آدرس پردازه مورد استفاده در آن ذخیره می‌شود. سپس با فراخوانی تابع allocproc فضای مورد نیاز از حافظه allocate شده و متغیر initproc برابر با اولین پردازه مقداردهی می‌شود. پس از آن تابع uvmfirst اجرا شده و یک پیج از حافظه به کاربر اختصاص می‌یابد (در واقع یک پیج از حافظه به پردازه داده می‌شود). در ادامه مقادیر اشاره‌گرهای برنامه و استک برابر صفر قرار داده می‌شوند. سپس پردازه نام‌گذاری شده و محل کار آن برابر با root سیستم قرار داده می‌شود. پس از آن نیز وضعیت پردازه به در حال اجرا تغییر داده می‌شود تا وارد روند اجرا شود. در انتها لاک مربوطه release می‌شود و فرایند پایان می‌یابد.

**سوال دوم**

تابع syscall که در فایل syscall.c قرار دارد، وظیفه دریافت و اجرای فراخوانی‌های سیستمی را دارد که به شکل entry به usys.pl افزوده شده‌اند. در هر بار اجرای این تابع، اطلاعات فراخوانی شامل نام و شماره آن (که پیش‌تر در فایل syscall.h ثبت شده) در رجیستر a7 قرار می‌گیرد. سپس با توجه به این اطلاعات، تابع مربوط به فراخوانی که پیش‌تر در فایل sysproc.c ذخیره شده اجرا می‌شود که خود آن نیز با جست‌وجو و اجرای تابعی که در فایل defs.h معرفی کرده‌ایم، تابع اصلی را از فایل proc.c فراخوانی کرده و خروجی آن را به عنوان خروجی فراخوانی سیستمی بازگردانی می‌کند.