

# Formativa I – strace

## Aluno: Fernando Alonso Piroga da Silva – Ciência da Computação

1. Quais chamadas de sistema foram acionadas e o que cada uma delas faz no SO?

- . execve – Executa um programa;
- . brk – Muda a quantidade de espaço alocada para o segmento de dado de chamada de processo;
- . arch\_prctl – Uma chamada de sistema do Linux que permite que um processo defina ou obtenha o estado de thread específico da arquitetura;
- . mmap – Mapeia páginas da memória;
- . access – Checa as permissões do usuário para um arquivo;
- . openat – Abre um arquivo usando o descritor de arquivo de diretório especificado como ponto de partida para a busca do caminho;
- . newfstatat – Obtém informações sobre um arquivo usando o descritor de arquivo de diretório especificado como ponto de referência para a localização do caminho;
- . close – Fecha um descritor de arquivo;
- . read – Lê de um descritor de arquivo;
- . pread64 – Lê de um descritor de arquivo em desvio de dado;
- . mprotect – Define a proteção do mapeamento de memória;
- . munmap - Desmapeia páginas de memória;
- . geteuid – Pega o ID do usuário efetivo;
- . lseek - Reposicionar deslocamento de arquivo de leitura/gravação;
- . write – Escreve em um descritor de arquivo;

Fonte: <https://www.ime.usp.br/~kon/MAC211/syscalls.html>

2. Por que os comandos 'whoami' e 'echo' precisam de chamadas de sistema?

R. Eles precisam interagir com o kernel para cumprir o seu objetivo, o whoami, para imprimir o nome do usuário, precisa acessar o user id do processo em execução e esse id é mantido pelo kernel, logo ele precisa de várias chamadas de sistema para adquirir esse dado. Já o echo, para imprimir o texto, precisa escrever na saída padrão (stdout), que é um descritor de arquivo ligado ao terminal que por sua vez é um recurso do sistema operacional, logo precisa de chamadas de sistema para poder executar essa ação.