



Linux System Programming & Kernel Interface



life.augmented

TABLE OF CONTENT



TRAINING PREVIEW



Intro to Linux System
Programming



Linux System
Programming



Advanced Linux System
Programming & Kernel
Interface

BACK

NEXT



life.augmented

Training Preview

- The Training takes about 7 Months of totally 37 sessions, 185 training hours and 2 projects.
- It started at 01/2024 and ended just before 09/2024.
- It was taken at STMicroelectronics company Egypt branch.

BACK

NEXT



life.augmented

What is STM?

- STMicroelectronics is a major European multinational semiconductor company that was established in 1987 following the merger of two state-owned entities: SGS Microelettronica of Italy and Thomson Semiconducteurs of France. Initially named SGS-THOMSON, the company adopted the name STMicroelectronics in 1998. Headquartered in Plan-les-Ouates, Switzerland, ST is the largest semiconductor company in Europe and one of the top players globally.
- STMicroelectronics designs, manufactures, and markets a wide array of semiconductor products, including microcontrollers, power management chips, sensors, and integrated circuits. The company is heavily involved in automotive, industrial, consumer electronics, and communication technologies, and has a strong presence in emerging fields like silicon carbide for electric vehicles.

BACK

NEXT



life.augmented

WHAT IS STM?



T1 - Introduction to LSP

- This training provides an introduction to Linux system programming, covering the history of Linux, some basic commands, system calls, processes, I/O redirection, environment variables and basics of Linux file systems.
- It takes totally 10 sessions with 50 Hours and a final project.

BACK

NEXT



life.augmented

T1 - Introduction to LSP

TABLE OF CONTENTS

- Linux History
- System Calls
- The exec Family of System Calls
- Forking Processes
- I/O Redirection in Linux
- Environment Variables
- File Systems Introduction

BACK

NEXT



life.augmented

T1 - Introduction to LSP

TABLE OF CONTENTS

- Linux History
- System Calls
- The exec Family of System Calls
- Forking Processes
- I/O Redirection in Linux
- Environment Variables
- File Systems Introduction

BACK

NEXT



life.augmented

T1 - Introduction to LSP

TABLE OF CONTENTS

- Linux History
- System Calls
- The exec Family of System Calls
- Forking Processes
- I/O Redirection in Linux
- Environment Variables
- File Systems Introduction

BACK

NEXT



life.augmented

T1 - Introduction to LSP

TABLE OF CONTENTS

- Linux History
- System Calls
- The exec Family of System Calls
- Forking Processes
- I/O Redirection in Linux
- Environment Variables
- File Systems Introduction

BACK

NEXT



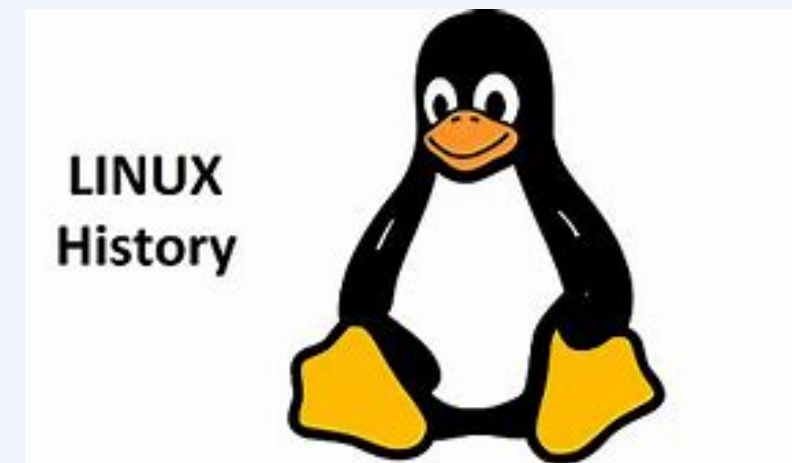
life.augmented

T1 - Introduction to LSP

LINUX HISTORY

- The first UNIX operating system was developed in 1969 by Ken Thompson at AT&T's Bell Laboratories.
- In 1970, UNIX was rewritten for the PDP-11 mini-computer.
- In 1984, Richard Stallman initiated the GNU Project to create a 'free' UNIX-like operating system.
- In 1991, Linus Torvalds released the first version of the Linux kernel.

BACK



life.augmented

T1 - Introduction to LSP

LINUX TERMINAL

- Then we started getting our hands dirty with the linux structure and the terminal by learning some commands.



BACK

NEXT

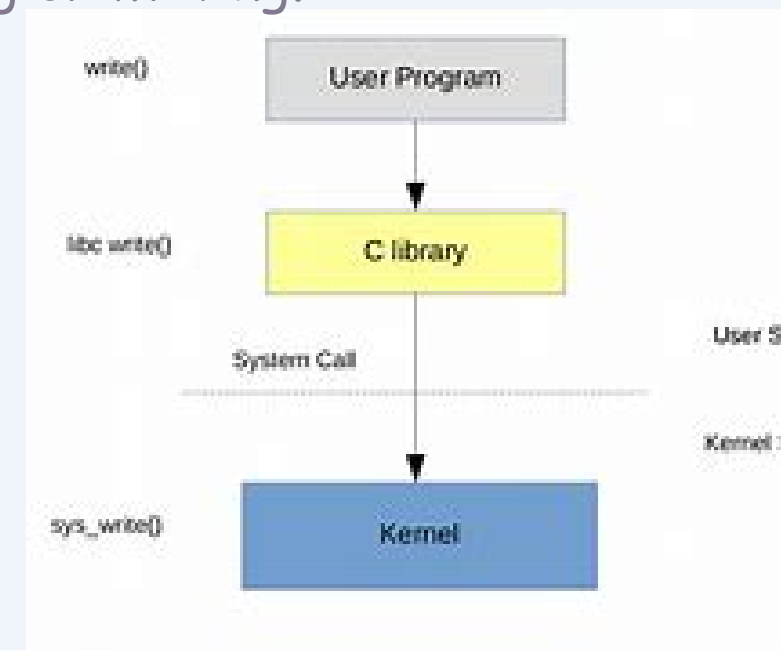


life.augmented

T1 - Introduction to LSP

SYSTEM CALLS

- System calls provide the interface between user applications and the kernel.
- Common system calls include open, read, write, and close for file management.
- Example: open() is used to open a file or device for reading or writing.



BACK

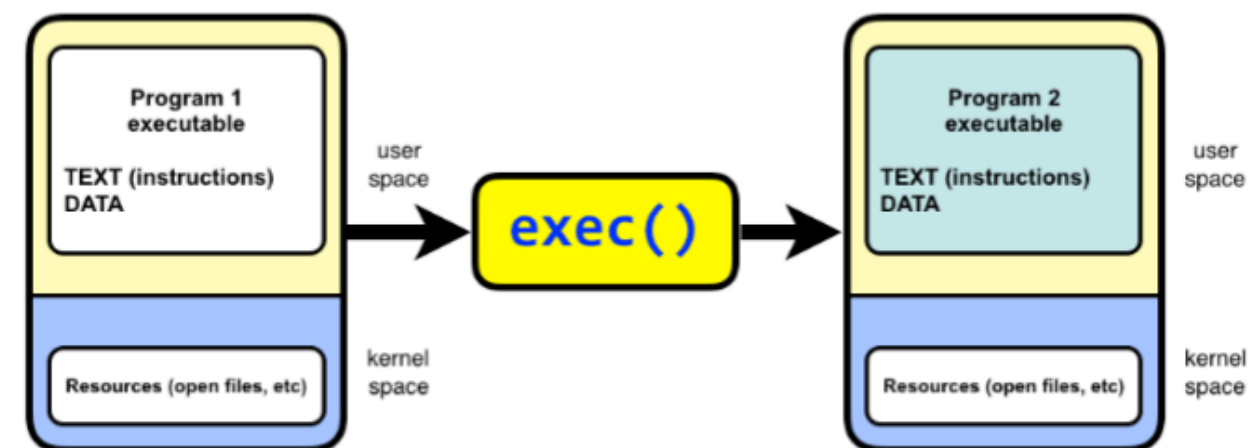


life.augmented

T1 - Introduction to LSP

PROCESS CREATION - EXEC

- Exec family allows a running process to replace its own image with a new program.
- Commonly used calls: `execv` and `execvp`.
- Exec calls help in running new programs while passing arguments and handling file descriptors.



BACK

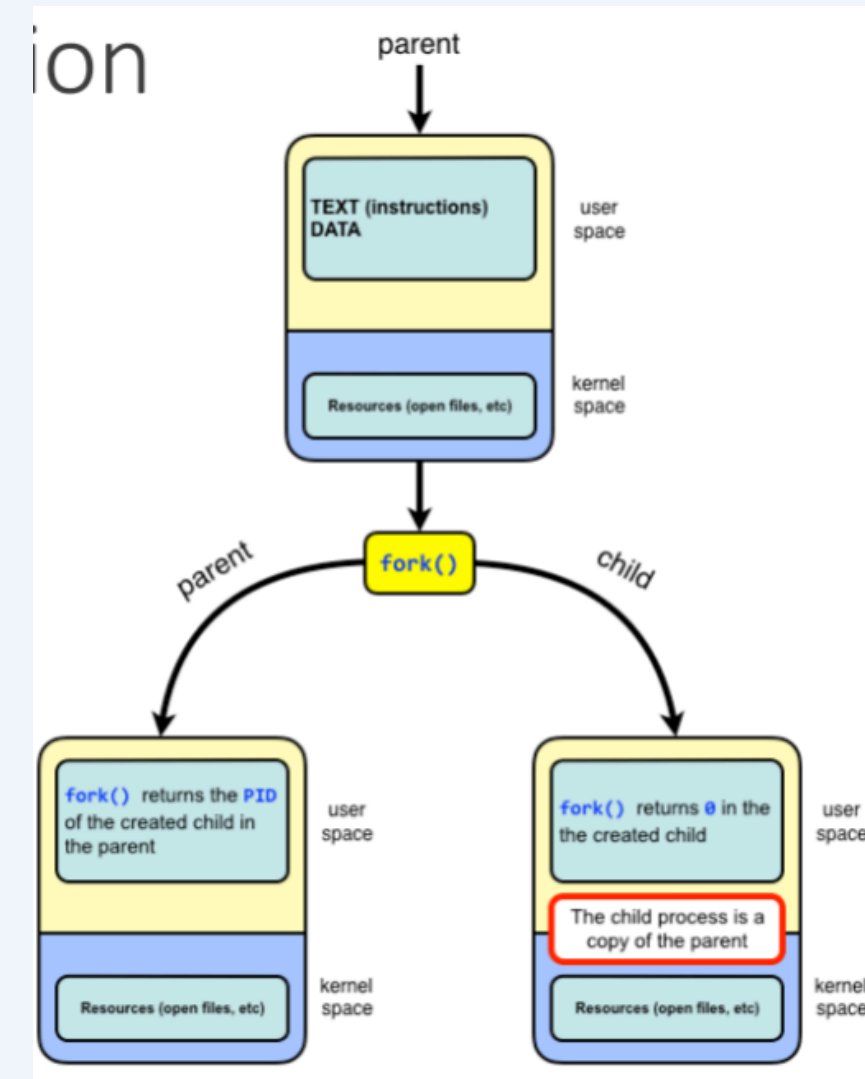


life.augmented

T1 - Introduction to LSP

PROCESS CREATION - FORK

- `fork()` is used to create a new process by duplicating the calling process.
- The new process is called the child process.
- The parent process can manage the child process using `wait()` and `waitpid()`.



BACK

NEXT

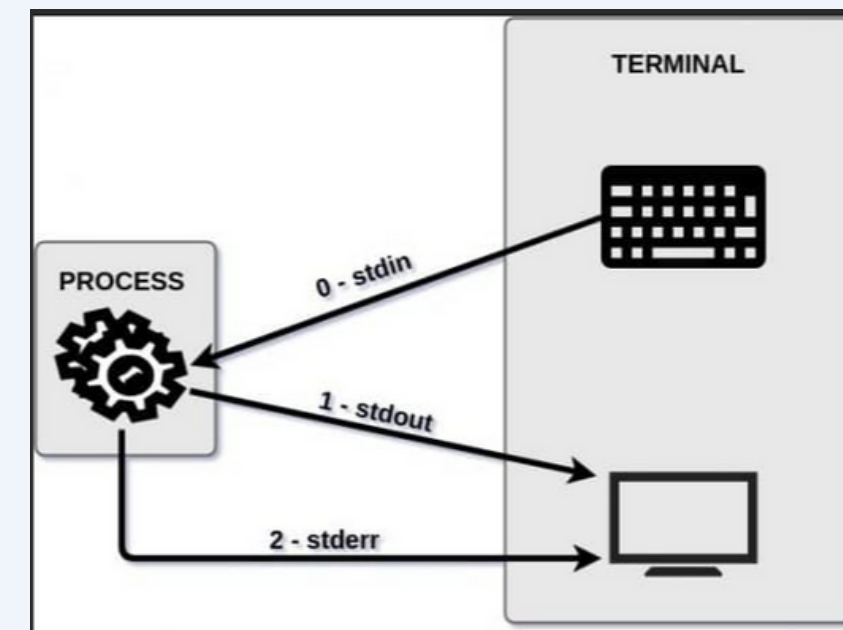


life.augmented

T1 - Introduction to LSP

I/O REDIRECTION IN LINUX

- I/O redirection allows the input and output of commands to be redirected.
- Types of redirection include standard output (>), standard input (<), and standard error (2>).
- **Example:** `echo 'Hello' > file.txt` redirects output to a file.



BACK



life.augmented

T1 - Introduction to LSP

ENVIRONMENT VARIABLES

- Environment variables store configuration settings for the system and processes.
- `$PATH` specifies directories where executable programs are located.
- To set an environment variable, use: `export VARIABLE_NAME='value'`.



BACK



life.augmented

T1 - Introduction to LSP

PROJECT - EZZAT SHELL

- implement a basic shell that allows users to execute commands. It handles both built-in commands (like echo, pwd, and cd) and external commands (using `execvp`). The shell creates a new process for each command entered by the user, using `fork`, and the parent process waits for the child to finish before continuing. It provides an interactive prompt, processes the user's input, and allows for command execution in a loop until the user types exit.

BACK

NEXT



life.augmented



THANK YOU

by Mohamed Ezzat



life.augmented