Session 19

the Linux system calls: execl(), execlp(), dup(), and dup2() — all commonly used in process control and file descriptor management.

1. execl() – Execute a program

Purpose:

Replaces the current process image with a new process (runs another program).

Syntax:

#include <unistd.h>

int execl(const char *path, const char *arg0, ..., NULL);

path: Full path to the executable.

arg0: Name of the program (usually same as filename).

arg1...argN: Arguments passed to the program.

Must be NULL-terminated.

Return Value:

Does not return on success.

Returns -1 on error (if execution fails).

Example:

execl("/bin/ls", "ls", "-l", NULL);

Replaces current process with ls -l command.

2. execlp() – Execute using PATH environment variable

Purpose:

Like execl(), but searches the PATH environment variable to locate the executable.

Syntax:

#include <unistd.h>

int execlp(const char *file, const char *arg0, ..., NULL);

file: Name of the executable (no need to give full path).

Example:

execlp("ls", "ls", "-a", NULL);

Searches for ls in directories listed in $PATH.

Difference: execl() vs execlp()

Feature execl() execlp()

Path required Full path to program Only program name (uses PATH)

PATH lookup No Yes

3. dup() – Duplicate a file descriptor

Purpose:

Duplicates an existing file descriptor to the lowest available unused file descriptor.

Syntax:

#include <unistd.h>

int dup(int oldfd);

Return Value:

New file descriptor (≥ 0) on success.

-1 on error.

Example:

int fd1 = open("file.txt", O_RDONLY);

int fd2 = dup(fd1); // fd2 now refers to the same file as fd1

4. dup2() – Duplicate to a specific file descriptor

Purpose:

Duplicates a file descriptor to a specified value.

Syntax:

#include <unistd.h>

int dup2(int oldfd, int newfd);

If newfd is already open, it is closed first.

If oldfd == newfd, dup2() does nothing and returns newfd.

Example:

dup2(fd1, 1); // Redirects stdout to file opened in fd1

Difference: dup() vs dup2()

Feature	**dup()**	**dup2()**
Target descriptor	Chooses lowest available	You specify newfd
Auto-close newfd?	N/A	✅ Yes (if already open)
Use case	Basic duplication	Redirection (stdin, stdout)

summary Table

System Call	**Purpose**	**Key Feature**
execl()	Execute program (absolute path)	Replaces current process
execlp()	Execute using PATH	Doesn't need full path
dup()	Duplicate file descriptor	Returns lowest available descriptor
dup2()	Duplicate to specific descriptor	Allows redirection