



BAHIR DAR UNIVERSITY
BAHIR DAR INSTITUTE OF TECHNOLOGY (BiT)
FACULTY OF computing
Operating system and system programming

Individual Assignment

Name: Tsehay Adugna

Section: A

ID Number :1602632

Title: System call

Submission date: 16/08/2017

Submitted to: Lec Wendemu B

Pivot_root()

system call is a programmatic way in which a computer program requests a service from the kernel of the operating system it is executed on.

`pivot_root()` is a system call that changes the root filesystem (i.e., `/`) of the current process to a new location. It's commonly used in container technologies, `initramfs` scripts, or custom boot environments.

The `pivot_root` command in Linux is used to change the root file system of a running system. It is primarily used during the boot process or in specialized scenarios like containerization (e.g., Docker, LXC) to switch from an initial temporary root file system (often an `initramfs`) to the actual root file system on the disk.

`pivot_root` is a system call and command that allows the root file system (`/`) to be replaced with a new one.

It moves the current root file system to a different directory (e.g., `/oldroot`) and mounts the new root file system in its place.

After the switch, the old root file system can be unmounted or used for other purposes.

Why is Pivot_root used

1. During System Boot

When Linux boots, it often starts with a small, temporary root file system (`initramfs`) loaded into memory. This temporary file system contains essential tools and drivers needed to mount the actual root file system on the disk. Once the real root file system is ready, `pivot_root` is used to switch to it.

2.Containerization:

In container environments, `pivot_root` is used to isolate the container's file system from the host system. It allows the container to have its own root file system.

3.System Recovery

In recovery scenarios, `pivot_root` can be used to switch to a recovery file system for troubleshooting or repair.

How pivot_root Works

1. Mount the New Root

First, the new root file system is mounted at a temporary mount point, such as /newroot.

2. Call pivot_root

The pivot_root command is used to switch the root filesystem. It takes two arguments:

The path to the new root directory.

A directory under the new root where the old root will be moved (e.g., /newroot/oldroot).

3. Unmount the Old Root

Once the root has been switched, the old root (now accessible under /oldroot) can be unmounted to clean up and free resources.

Prototype

```
int pivot_root(const char *new_root, const char *put_old);
```

- new_root: The new root directory.
- put_old: A location under the new root where the old root will be mounted (so it can later be unmounted if desired).

Example Use Case:

When booting a system using an initramfs, pivot_root is used to switch from the temporary initial root filesystem (like a RAM disk) to the real root filesystem (like the SD card or another partition).

Requirements:

- Both new_root and put_old must be on the same filesystem.
- pivot_root requires root privileges.
- Not all environments (like some containers) allow its usage due to limited capabilities.

Implementation of pivot_root() on Raspberry Pi:

This script assumes you're in a minimal boot environment, Such as an initramfs ora live rescue shell - not from a fully running Raspberry Pi OS. **WARNING** : Don't run this on your main system unless you're sure. Use a test setup or second SD card.

pivot_root.sh — Minimal Example Script

```
#!/bin/sh

# Mount essential filesystems
mount -t proc none /proc
mount -t sysfs none /sys
mount -t devtmpfs none /dev

# Mount new root filesystem (replace with actual device)
mount /dev/sda1 /mnt

# Prepare old_root directory
mkdir -p /mnt/old_root

# Move into new root
cd /mnt

# Do the pivot
pivot_root . old_root

# Unmount unnecessary mounts from old root
umount /old_root/proc
umount /old_root/sys
umount /old_root/dev

# Start the real init system
exec chroot . /sbin/init
```

Instructions:

1. Save as `pivot_root.sh`.
2. Make it executable: `chmod +x pivot_root.sh`.
3. Boot into a **minimal initramfs** or shell (like Alpine `initrd` or BusyBox shell).
4. Run the script as `root`.

There's a minimal C code example that calls the `pivot_root()` system call directly.

```
#define _GNU_SOURCE

#include <unistd.h>

#include <stdio.h>

#include <sys/syscall.h>

#include <errno.h>
```

```
int main() {  
    const char *new_root = "/newroot";  
    const char *put_old = "/newroot/old_root";  
  
    int ret = syscall(SYS_pivot_root, new_root, put_old);  
    if (ret == 0) {  
        printf("pivot_root successful.\n");  
    } else {  
        perror("pivot_root failed");  
    }  
  
    return 0;  
}
```