

Proyecto 1: Boot Process of an Embedded System, its BSP and Development Environment

Nombre: Ing. Andrés Gómez Jiménez		Carne: 200935203
Nombre: Ing. Randy Céspedes Deliyore		Carne: 201054417

1. Preguntas Teóricas

Se comenta a continuación los resultados encontrados para las preguntas planteadas.

- ¿Qué es Device Tree y por qué es importante?

De acuerdo con [1], device tree (DT) es una estructura de datos para describir hardware no descubrible en un sistema computacional. Esta estructura de datos describe el hardware en el sistema (CPU, memoria, periféricos, etc) y como interactuar con estos. Se crea y es importante por que así no es necesario contener esta información de manera fija en el sistema operativo (OS), sino que las rutinas de arranque de más bajo nivel cargan esta información del ROM en el hardware y lo transfieren al cargar el OS.

- ¿Qué es un overlay de Device Tree y qué puede controlar?

De acuerdo con [2], el overlay permite generar expansión sobre un DT base existente. Por ejemplo, se puede crear un DT base que describe los componentes principales del sistema, pero con la capacidad de ser expandido al añadir nuevos componentes diversos. Se utiliza principalmente para añadir periféricos o nuevas interfaces de E/S.

- Explicar que son los fuses del procesador y los bootstrap resistors. Mencionar lo que estos controlan. Mencionar ejemplo de procesador en el que se controla el proceso de arranque con ambos métodos.

Estos componentes electrónicos suelen ser utilizados por el componente de arranque de más bajo nivel (como un BootROM) para determinar la localización del Second Stage Bootloader. En algunos casos, estos también pueden configurar si se debe validar la integridad de este componente a cargar y como hacerlo [3]. No fue posible encontrar un ejemplo específico de procesador comercial que formalmente documente la habilidad con ambos mecanismos.

- Investigar la localización en RAM del Device Tree en el RPI4 y las direcciones de RAM reservadas para la operación de U-Boot.

No se encontró documentación formal con respuestas certeras para esta pregunta. Algunos foros indican que el U-Boot se cargaría en la sección final de la RAM y el tamaño final dependerá de las configuraciones del mismo y la tarjeta de desarrollo.

- Relacionado con NFS. Investigar si el sistema de archivos completo se monta en RAM o cual es el criterio utilizado por el Kernel de Linux para saber qué cargar del sistema de archivos. Comparar con Initrd.

2. Resultados

Resultados aquí

2.1. Hardware utilizado

Para la realización de este proyecto se utilizó una RaspberryPi 4 con un **Canakit** como se puede en la figura 1.



Figura 1: Asignación de Recursos Máquina Virtual Ubuntu 18.04 en VirtualBox 6.1 [4].

2.2. Configuración Máquina Virtual

Se realiza la configuración de una máquina virtual utilizando el ambiente de virtualización de Oracle VirtualBox 6.1. Se utilizó la imagen creada por los profesores y se asignaron los recursos visibles en la figura 2 para poder realizar la compilación de los siguientes elementos:

- Bootloader de segunda etapa U-Boot.
- Kernel de Linux personalizado.
- File System personalizado con Yocto.
- Meta Layer con aplicación personalizada.

En el cuadro 1 hay un resumen de los sistemas utilizados.

Para conectar los sistemas del cuadro 1 se creó una red local con la configuración del cuadro 2.

Para realizar la posterior comunicación con el RPI4 se contó con un servidor TFT (visible en la figura 3) y un servidor NFS (visible en la figura 4).

2.3. Compilación Bootloader de segunda etapa U-Boot

Se realizó la compilación exitosa del bootloader de segunda etapa U-Boot como se puede ver en la figura 5. Esta se utilizó para la configuración de diversas variables de ambiente en el RPI4.

Cuadro 1: Resumen de sistemas utilizados en el proyecto

Nombre del Sistema	Características
	Sistema Operativo: Ubuntu 20.04 LTS
Computadora huesped	Procesador: Intel (T) Core(TM) i7-3770 Octo-core @3.4 GHz Memoria: DDR3 1600 MHz 16 GB
	Sistema Operativo: Ubuntu 18.04 LTS
Computadora virtualizada	Procesador: 4 cores asignados. Memoria: 11 GB RAM asignados.
	Sistema Operativo: Personalizado
RaspberryPI 4	Procesador: Broadcom, Quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5 GHz Memoria: 4 GB

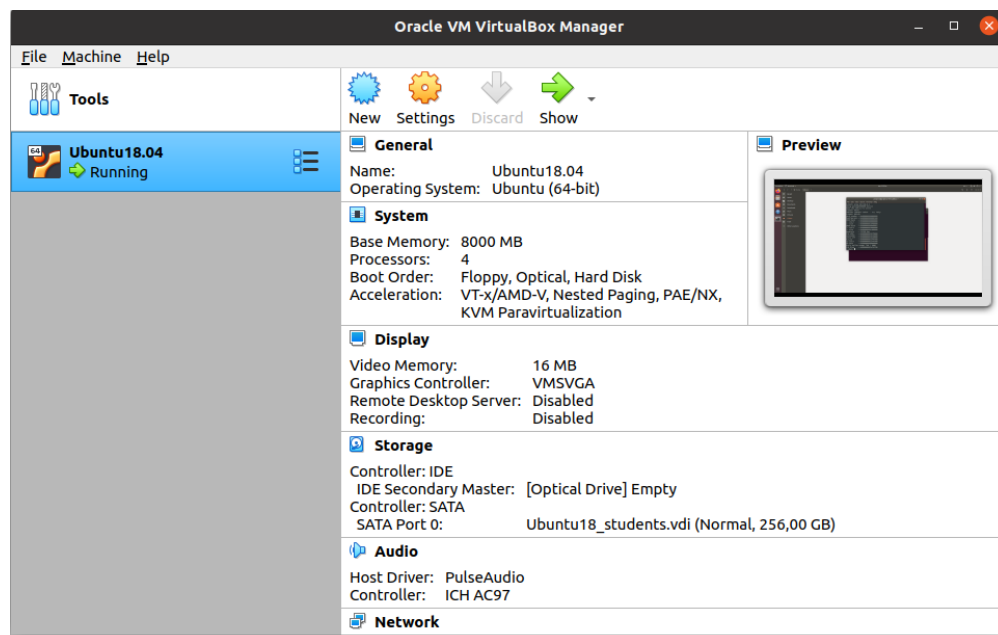


Figura 2: Asignación de Recursos Máquina Virtual Ubuntu 18.04 en VirtualBox 6.1

Cuadro 2: Resumen de sistemas utilizados en el proyecto

Nombre del Sistema	IP estática	Macara de subred	Gateway
Computadora huesped	192.168.8.1	255.255.255.0	-
Computadora virtualizada	192.168.8.2	255.255.255.0	-
RaspberryPI 4	192.168.8.3	255.255.255.0	-

2.4. Compilación Kernel

Se compiló un kernel personalizado para el RPI4 desabilitando el acceso llamado MMC/SD/SDIO card support. En la figura 6 se puede ver los resultados de la compilación.

Se utilizó el comando *bdinfo* para ver la información de las ubicaciones de memoria desde la consola de U-Boot como se puede ver en la figura 7.

Se escogió la dirección 0x8000 para cargar el kernel compilado, mismo que como se puede ver en la figura 8 tiene

```
project2@project2-VirtualBox: ~
File Edit View Search Terminal Help
project2@project2-VirtualBox:~$ sudo service tftpd-hpa status
● tftpd-hpa.service - LSB: HPA's tftp server
   Loaded: loaded (/etc/init.d/tftpd-hpa; generated)
   Active: active (running) since Tue 2020-06-16 19:52:10 CST; 3h 16min ago
     Docs: man:systemd-sysv-generator(8)
   Process: 1021 ExecStart=/etc/init.d/tftpd-hpa start (code=exited, status=0/SUC
   Tasks: 1 (limit: 4915)
    CGroup: /system.slice/tftpd-hpa.service
            └─1044 /usr/sbin/in.tftpd --listen --user tftp --address :69 --secure

jun 16 19:52:10 project2-VirtualBox systemd[1]: Starting LSB: HPA's tftp server.
jun 16 19:52:10 project2-VirtualBox tftpd-hpa[1021]: * Starting HPA's tftpd in.
jun 16 19:52:10 project2-VirtualBox tftpd-hpa[1021]: ...done.
jun 16 19:52:10 project2-VirtualBox systemd[1]: Started LSB: HPA's tftp server.
lines 1-13/13 (END)
```

Figura 3: Estado del servidor TFTP en la máquina virtual

```
project2@project2-VirtualBox: ~
File Edit View Search Terminal Help
project2@project2-VirtualBox:~$ sudo service nfs-kernel-server status
● nfs-server.service - NFS server and services
   Loaded: loaded (/lib/systemd/system/nfs-server.service; enabled; vendor prese
   Active: active (exited) since Tue 2020-06-16 19:52:08 CST; 3h 17min ago
   Process: 909 ExecStart=/usr/sbin/rpc.nfsd $RPCNFSDARGS (code=exited, status=0/
   Process: 906 ExecStartPre=/usr/sbin/exportfs -r (code=exited, status=0/SUCCESS
   Main PID: 909 (code=exited, status=0/SUCCESS)

jun 16 19:52:06 project2-VirtualBox systemd[1]: Starting NFS server and services
jun 16 19:52:08 project2-VirtualBox systemd[1]: Started NFS server and services.
lines 1-9/9 (END)
```

Figura 4: Estado del servidor NFS en la máquina virtual

un tamaño de 16 MB.

2.5. Compilación File System Yocto

Se realizó la compilación del File System para el RPI4 de forma exitosa usando Yocto como se puede ver en la figura 9.

2.6. Creación de aplicación de Integración Trapezoidal Compuesta en Octave

Se crea el código visible a continuación para resolver el problema de la Integración Trapezoidal Compuesta en el programa GNU Octave.

```
# Course: MP-6171 High Performance Embedded Systems
# Tecnológico de Costa Rica (www.tec.ac.cr)
# Developers Name: Andres Gomez (agomez10010@gmail.com) and Randy Cespedes
# (rcespedes27dds@gmail.com)
# This script is structured in Octave (https://www.gnu.org/software/octave/)
```

```
project2@project2-VirtualBox: ~
File Edit View Search Terminal Help

Welcome to minicom 2.7.1

OPTIONS: I18n
Compiled on Aug 13 2017, 15:25:34.
Port /dev/ttyUSB0, 23:11:36

Press CTRL-A Z for help on special keys

U-Boot 2020.04-rc3-dirty (Jun 07 2020 - 23:11:36 -0600)

DRAM:  3.9 GiB
RPI 4 Model B (0xc03111)
MMC:  mmcnr@7e300000: 1, emmc2@7e340000: 0
Loading Environment from FAT... OK
In:     serial
Out:    serial
Err:    serial
Net:    eth0: genet@7d580000
Hit any key to stop autoboot:  0
U-Boot>
```

Figura 5: Utilización exitosa del Bootloader de segunda etapa U-Boot

```
project2@project2-VirtualBox: ~/project2/git/linux
File Edit View Search Terminal Help

KSYM  .tmp_vmlinux.kallsyms1.o
LD     .tmp_vmlinux.kallsyms2
KSYM  .tmp_vmlinux.kallsyms2.o
LD     vmlinux
SORTTAB vmlinux
SYSMAP System.map
MODPOST 1610 modules
OBJCOPY arch/arm64/boot/Image
GZIP     arch/arm64/boot/Image.gz
CC [M]  drivers/net/wireless/broadcom/brcm80211/brcmfmac/brcmfmac.mod.o
LD [M]  kernel/configs.ko
CC [M]  drivers/bluetooth/btmrvl_sdio.mod.o
CC [M]  drivers/mmc/host/mmc_spi.mod.o
CC [M]  drivers/mmc/host/of_mmc_spi.mod.o
CC [M]  drivers/net/wireless/marvell/libertas/libertas_sdio.mod.o
CC [M]  drivers/net/wireless/marvell/mwifiex/mwifiex_sdio.mod.o
LD [M]  drivers/net/wireless/broadcom/brcm80211/brcmfmac/brcmfmac.ko
LD [M]  drivers/mmc/host/of_mmc_spi.ko
LD [M]  drivers/net/wireless/marvell/libertas/libertas_sdio.ko
LD [M]  drivers/bluetooth/btmrvl_sdio.ko
LD [M]  drivers/mmc/host/mmc_spi.ko
LD [M]  drivers/net/wireless/marvell/mwifiex/mwifiex_sdio.ko
make[1]: Leaving directory '/home/project2/project2/git/linux/our_binaries'
project2@project2-VirtualBox:~/project2/git/linux$
```

Figura 6: Resultados de la compilación del Kernel

```
# General purpose: Composite Trapezoidal Rule Prototyping
# Input: Lower value, upper value and sub-intervals
# Output: numerical approximation

# Define constant values for test case 1
a1 = 0;
b1 = 10;
```

```
U-Boot> bdfinfo
arch_number = 0x0000000000000000
boot_params = 0x0000000000000100
DRAM bank = 0x0000000000000000
-> start = 0x0000000000000000
-> size = 0x000000003b400000
DRAM bank = 0x0000000000000001
-> start = 0x0000000040000000
-> size = 0x00000000bc000000
baudrate = 115200 bps
TLB addr = 0x000000003b3f0000
relocaddr = 0x000000003b359000
reloc off = 0x000000003b2d9000
irq_sp = 0x000000003af49780
sp_start = 0x000000003af49780
FB base = 0x0000000000000000
Early malloc usage: 790 / 2000
fdt_blob = 0x000000003af49790
U-Boot>
```

Figura 7: Información de memoria RPI4 por medio del comando *bdfinfo* en la consola de U-Boot

```
project2@project2-VirtualBox: ~/project2/git/linux
File Edit View Search Terminal Help
project2@project2-VirtualBox:~/project2/git/linux$ ls our_binaries/arch/arm64/boot -lh
total 23M
drwxr-xr-x 32 project2 project2 4,0K jun  9 17:53 dts
-rwxr-xr-x  1 project2 project2 16M jun 16 23:26 Image
-rw-r--r--  1 project2 project2 6,8M jun 16 23:27 Image.gz
project2@project2-VirtualBox:~/project2/git/linux$
```

Figura 8: Imagen compilada del kernel personalizado para RPI4

```
project2@project2-VirtualBox:~/yocto/poky/build$ bitbake core-image-base
Loading cache: 100% |#####| Time: 0:00:00
Loaded 1329 entries from dependency cache.
NOTE: Resolving any missing task queue dependencies

Build Configuration:
BB_VERSION      = "1.44.0"
BUILD_SYS       = "x86_64-linux"
NATIVELSBSTRING = "universal"
TARGET_SYS      = "aarch64-poky-linux"
MACHINE         = "raspberrypi4-64"
DISTRO          = "poky"
DISTRO_VERSION  = "3.0.3"
TUNE_FEATURES   = "aarch64 cortexa72 crc crypto"
TARGET_FPU      = ""
meta
meta-poky
meta-yocto-bsp   = "zeus:ca9dd4b8eab400f7364f522b1383c21bf47351a"
meta-raspberrypi = "zeus:0e05098853eea77032bf9cf81955679edd2f35d"

Installing tasks: 100% |#####| Time: 0:00:02
#####
State summary: Wanted 1441 Found 67 Missed 1374 Current 0 (4% match, 0% complete)
NOTE: Executing Tasks
NOTE: Setscene tasks completed
WARNING: libxext-1.3.4-r0 do_fetch: Failed to fetch URL https://www.x.org/releases/individual/lib/libxext-1.3.4.tar.bz2, attempting MIRRORS if available
WARNING: libxrender-0.9.10-r0 do_fetch: Failed to fetch URL https://www.x.org/releases/individual/lib/libxrender-0.9.10.tar.bz2, attempting MIRRORS if available
WARNING: libfixes-5.0.3-r0 do_fetch: Failed to fetch URL https://www.x.org/releases/individual/lib/libfixes-5.0.3.tar.bz2, attempting MIRRORS if available
WARNING: libxrandr-1.5.2-r0 do_fetch: Failed to fetch URL https://www.x.org/releases/individual/lib/libxrandr-1.5.2.tar.bz2, attempting MIRRORS if available
WARNING: libp11access-0.16-r0 do_fetch: Failed to fetch URL https://www.x.org/releases/individual/lib/libp11access-0.16.tar.bz2, attempting MIRRORS if available
WARNING: libxshmfence-1.3-r0 do_fetch: Failed to fetch URL https://www.x.org/releases/individual/lib/libxshmfence-1.3.tar.bz2, attempting MIRRORS if available
WARNING: xrandr-1.5.1-r0 do_fetch: Failed to fetch URL https://www.x.org/releases/individual/app/xrandr-1.5.1.tar.xz, attempting MIRRORS if available
WARNING: libdrm-2.4.99-r0 do_fetch: Failed to fetch URL https://dri.freedesktop.org/libdrm/libdrm-2.4.99.tar.bz2, attempting MIRRORS if available
WARNING: libxfont-1.1.4-r0 do_fetch: Failed to fetch URL https://www.x.org/releases/individual/lib/libxfont-1.1.4.tar.bz2, attempting MIRRORS if available
WARNING: libdmg-1.1.3-r0 do_fetch: Failed to fetch URL https://www.x.org/releases/individual/lib/libdmg-1.1.3.tar.bz2, attempting MIRRORS if available
WARNING: nakedepend-native-1.0.6-r0 do_fetch: Failed to fetch URL https://www.x.org/releases/individual/util/nakedepend-1.0.6.tar.gz, attempting MIRRORS if available
WARNING: pixman-0.38.4-r0 do_fetch: Failed to fetch URL https://www.x.org/releases/individual/lib/pixman-0.38.4.tar.bz2, attempting MIRRORS if available
WARNING: fontconfig-2.13.1-r0 do_fetch: Failed to fetch URL https://fontconfig.org/release/fontconfig-2.13.1.tar.gz, attempting MIRRORS if available
WARNING: mesa-21.0.6-r0 do_fetch: Failed to fetch URL https://mesa.freedesktop.org/archive/mesa-21.0.6.tar.xz, attempting MIRRORS if available
WARNING: cairo-1.16.0-r0 do_fetch: Failed to fetch URL https://cairographics.org/releases/cairo-1.16.0.tar.xz, attempting MIRRORS if available
NOTE: Tasks Summary: Attempted 4086 tasks of which 316 didn't need to be rerun and all succeeded.

Summary: There were 15 WARNING messages shown.
project2@project2-VirtualBox:~/yocto/poky/build$
```

Figura 9: Resultado de compilación del File System de Yocto para RPI4

```
n1 = 200;
expected_result1 = 1.47112726584903;

# Define constant values for test case 2
a2 = 3.5;
b2 = 45;
```



```
n2 = 150;
expected_result2 = 0.256335035737283;

# **** Define the functions to use ****

# This function calculates the value of  $f(x) = y = 1 / (1 + x^2)$ 
function y = f(x)
    y = 1 / (1 + x*x);
end

# This function calculates the integral of a function using the composite trapezoidal rule de
function integral = Trapezoidal(lower, upper, subInterval)
    # Start by calculating h
    h = (upper - lower) / subInterval;

    # Calculate initial value for result
    integral = f(lower) + f(upper);

    # Now add the iterative portion
    sum = 0;
    for j = 1:(subInterval-1)
        sum = sum + f(lower + (j*h));
    end

    # Calculate final value for integral result
    integral = h*(integral + 2*sum)/2;
end

# **** Exercise function with test values here and print results ****

integral_result1 = Trapezoidal(a1, b1, n1);
fprintf("Test 1 >> Integral Result: %.10f. Expected Result: %.10f\n\n", integral_result1, exp

integral_result2 = Trapezoidal(a2, b2, n2);
fprintf("Test 2 >> Integral Result: %.10f. Expected Result: %.10f\n\n", integral_result2, exp
```

2.7. Creación de aplicación de Integración Trapezoidal Compuesta en C

Se procedió a crear el código C para la implementación en Octave de la sección anterior. El código resultante se puede ver a continuación.

```
/*
=====
Name       : trapezoidal.c
Author      : agomez and rcespedes
Version     : 1.0.0
Description : General purpose: Composite Trapezoidal Rule Application
Course:     MP-6171 High Performance Embedded Systems
Tecnologico de Costa Rica (www.tec.ac.cr)
Input:      Lower value, upper value and sub-intervals
```

Output: numerical approximation

```
*/

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

// Functions declaration
void check_required_inputs(int l_flag, int u_flag, int n_flag, char *prog_name);
void check_inputs_values(float lower, float upper, int n);
float Trapezoidal(float lower, float upper, int subInterval);
float f(float x);
void testing(void);

// Constant for help on usage
static char usage[] = "usage: %s [-h] [-a] -l LowerValue -u UpperValue -n
subIntervalsValue\n"
    "-l LowerValue specifies lower value.\n"
    "-u UpperValue specifies upper value.\n"
    "-n subIntervalsValue specifies the sub-intervals value.\n"
    "-a displays the information of the authors of the program.\n"
    "-h displays the usage message to let the user know how to execute
the application.\n";

int main(int argc, char *argv[]) {
    // Enable line below for testing
    //testing();

    // Define required parameters for calculation
    float lower;
    float upper;
    int n;

    // Use flags below to tell if the required arguments were provided
    int l_flag = 0;
    int u_flag = 0;
    int n_flag = 0;

    // Parse command line parameters
    int c;
    while ((c = getopt (argc, argv, "hal:u:n:")) != -1)
        switch (c){
            case 'l':
                l_flag = 1;
                lower = atof(optarg);
                break;
            case 'u':
                u_flag = 1;
                upper = atof(optarg);
                break;
```



```
        case 'n':
            n_flag = 1;
            n = atoi(optarg);
            break;
        case 'h':
            fprintf(stderr, usage, argv[0]);
            exit(1);
            break;
        case 'a':
            printf("Authors: agomez and rcespedes\n");
            exit(1);
            break;
        case ':':
            break;
        case '?':
            fprintf(stderr, "Unknown option '-%c'.\n", optopt);
            return 1;
        default:
            abort();
    }
    // Check that required inputs were provided
    check_required_inputs(l_flag, u_flag, n_flag, argv[0]);
    // Verify inputs have values that make sense
    check_inputs_values(lower, upper, n);

    // Now we are ready to do calculation
    puts("Starting calculation...");
    float result = Trapezoidal(lower, upper, n);
    printf("Integral Result is %.8f\n", result);

    return EXIT_SUCCESS;
}

void check_required_inputs(int l_flag, int u_flag, int n_flag, char *prog_name){
    // Check required arguments were provided. Print error and abort otherwise
    if (!l_flag){
        fprintf(stderr, "%s: missing -l option\n", prog_name);
        fprintf(stderr, usage, prog_name);
        exit(1);
    }
    if (!u_flag){
        fprintf(stderr, "%s: missing -u option\n", prog_name);
        fprintf(stderr, usage, prog_name);
        exit(1);
    }
    if (!n_flag){
        fprintf(stderr, "%s: missing -n option\n", prog_name);
        fprintf(stderr, usage, prog_name);
        exit(1);
    }
}
```

```
// This function checks that the input values meet certain conditions
void check_inputs_values(float lower, float upper, int n){
    // Confirm upper is greater than lower
    if(lower >= upper){
        printf("ERROR: UpperValue must be a number greater than
        LowerValue\n");
        exit(1);
    }
    // Confirm n is greater than 0
    if(n <= 0){
        printf("ERROR: subIntervalsValue must be a positive integer
        number\n");
        exit(1);
    }
}

// This function calculates the integral of a function using the composite
trapezoidal rule description
float Trapezoidal(float lower, float upper, int subInterval){
    // Start by calculating h
    float h = (upper - lower) / subInterval;

    // Calculate initial value for result
    float integral = f(lower) + f(upper);

    // Now add the iterative portion
    float sum = 0;
    for(int j=1; j<=(subInterval-1); j++){
        sum = sum + f(lower + (j*h));
    }

    // Calculate final value for integral result
    integral = h*(integral + 2*sum)/2;

    return integral;
}

// This function calculates the value of  $f(x) = y = 1 / (1 + x^2)$ 
float f(float x){
    float y = 1 / (1 + x*x);
    return y;
}

// Function used for testing purposes only
void testing(void){
    // Define constant values for test case 1
    float a = 0;
    float b = 10;
    int n = 200;
    float expected_result = 1.47112726584903;
    printf("Test 1 >> Integral Result: %.10f. Expected Result: %.10f\n\n",
    Trapezoidal(a, b, n), expected_result);
}
```

```
// Define constant values for test case 2
a = 3.5;
b = 45;
n = 150;
expected_result = 0.256335035737283;
printf("Test 2 >> Integral Result: %.10f. Expected Result: %.10f\n\n",
Trapezoidal(a, b, n), expected_result);
}
```

2.8. Meta Layer

Se utilizó el esquema de compilación autotools siguiendo el tutorial [5] y se compiló el binario de la aplicación de Integración Trapezoidal Compuesta en C. El resultado de correr en la aplicación en la máquina virtual se puede ver en la figura 10.

```
project2@project2-VirtualBox:~/Yocto/poky/meta-tec/recipes-trapezoidal/trapezoid
al/trapezoidal-1.0/src$ ./trapezoidal -l 0 -u 2 -n 100
Starting calculation...
Integral Result is 1.10714316
project2@project2-VirtualBox:~/Yocto/poky/meta-tec/recipes-trapezoidal/trapezoid
al/trapezoidal-1.0/src$
```

Figura 10: Resultado de correr el código compilado de la de la aplicación de Integración Trapezoidal Compuesta en C

En la figura 12 se puede ver una captura del árbol de la receta para el paquete trapezoidal-1.0. Y en la figura ?? se puede ver el resultado de incluir meta-tec dentro del file system de Yocto.

```
File Edit View Search Terminal Help
project2@project2-VirtualBox:~/Yocto/poky/meta-tec$ tree
├── conf
│   └── layer.conf
├── COPYING.MIT
├── README
├── recipes-tec
│   └── trapezoidal
│       ├── trapezoidal-1.0
│       │   └── trapezoidal-1.0.tar.gz
│       └── trapezoidal-1.0.bb
4 directories, 5 files
project2@project2-VirtualBox:~/Yocto/poky/meta-tec$
```

Figura 11: Resultado de comando `tree` para la receta de meta-tec que contiene el paquete trapezoidal-1.0

Finalmente se puede observar el resultado de correr el programa compilado en el File System del RPI4 en la figura 13.

```

project2@project2-VirtualBox: ~/Yocto/poky/build$ bitbake core-image-base
WARNING: Layer hello should set LAYERSERIES_COMPAT_hello in its conf/layer.conf file to list the core layer names it is compatible with.
WARNING: Layer tec should set LAYERSERIES_COMPAT_tec in its conf/layer.conf file to list the core layer names it is compatible with.
WARNING: Layer hello should set LAYERSERIES_COMPAT_hello in its conf/layer.conf file to list the core layer names it is compatible with.
WARNING: Layer tec should set LAYERSERIES_COMPAT_tec in its conf/layer.conf file to list the core layer names it is compatible with.
Parsing recipes: 100% |#####| Time: 0:00:35
Parsing of 803 .bb files complete (0 cached, 803 parsed). 1331 targets, 57 skipped, 0 masked, 0 errors.
NOTE: Resolving any missing task queue dependencies

Build Configuration:
BB_VERSION      = "1.44.0"
BUILD_SYS       = "x86_64-linux"
NATIVELSBSTRING = "universal"
TARGET_SYS      = "aarch64-poky-linux"
MACHINE         = "raspberrypi4-64"
DISTRO          = "poky"
KDISTRO_VERSION = "3.0.3"
TUNE_FEATURES   = "aarch64 cortexa72 crc crypto"
TARGET_FPU      = ""
meta
meta-poky
meta-yocto-bsp   = "zeus:ca9dd4b8eab40ef736a4f522b1383c21bf47351a"
meta-raspberrypi = "zeus:0e05098853eea77032bf9cf81955679edd2f35d"
meta-hello
meta-tec         = "zeus:ca9dd4b8eab40ef736a4f522b1383c21bf47351a"

Initialising tasks: 100% |#####| Time: 0:00:02
Sstate summary: Wanted 14 Found 0 Missed 14 Current 1439 (0% match, 99% complete)
NOTE: Executing Tasks
NOTE: Setscene tasks completed
NOTE: Tasks Summary: Attempted 4114 tasks of which 4077 didn't need to be rerun and all succeeded.

Summary: There were 4 WARNING messages shown.
project2@project2-VirtualBox: ~/Yocto/poky/build$

```

Figura 12: Resultado de compilación del File System de Yocto para RPI4 con el meta layer *meta-tec*

```

project2@project2-VirtualBox: ~
File Edit View Search Terminal Help

Starting syslogd/klogd: done
[ ok ]rtng Avahi mDNS/DNS-SD Daemon: avahi-daemon
Starting Telephony daemon
Starting Linux NFC daemon
Starting tcf-agent: OK
umount: /mnt/.psplash: not mounted.

Poky (Yocto Project Reference Distro) 3.0.3 raspberrypi4-64 ttyAMA0

raspberrypi4-64 login: root
root@raspberrypi4-64:~# hello
Hello world using Yocto
root@raspberrypi4-64:~# trapezoidal -h
usage: trapezoidal [-h] [-a] -l LowerValue -u UpperValue -n subIntervalsValue
-l LowerValue specifies lower value.
-u UpperValue specifies upper value.
-n subIntervalsValue specifies the sub-intervals value.
-a displays the information of the authors of the program.
-h displays the usage message to let the user know how to execute the applicatio
n.
root@raspberrypi4-64:~# trapezoidal -l 0 -u 2 -n 100
Starting calculation...
Integral Result is 1.10714316
root@raspberrypi4-64:~#

```

Figura 13: Corriendo trapezoidal en RPI4

Referencias

- [1] devicetree.org, *The DeviceTree Specification*, Online; accessed 15-June-2020, 2020. dirección: <https://www.devicetree.org/>.
- [2] raspberrypi.org, *Device Trees, overlays, and parameters*, Online; accessed 16-June-2020. dirección: <https://www.raspberrypi.org/documentation/configuration/device-tree.md>.
- [3] emergingdefense, *Bootrom versus Bootloader*, Online; accessed 17-June-2020. dirección: <https://www.emergingdefense.com/blog/2019/5/20/bootrom-versus-bootloader>.

- [4] Canakit, *Raspberry Pi 4 Starter Kit*, Online; accessed 17-June-2020, 2020. dirección: <https://www.canakit.com/raspberry-pi-4-starter-kit.html>.
- [5] S. Arriola, *Custom Meta Layer Yocto*, Online; accessed 10-June-2020, 2020. dirección: <https://www.youtube.com/watch?v=hgYXqSJcQU>.