

5. Gaia. Memoriaren kudeaketa

Ariketak / Laborategia

1. Ariketa. Konpilazio faseak gcc konpiladorearekin

1. Linuxeko *man* laguntza kontsultatuta, aztertu nola ikus daitekeen konpilazioaren fase desberdinak
2. Probatu **kopiatu.c** programaren konpilazioa
3. Konpilazioaren fase bakoitza komentatu
4. Saia zaitez beste konpiladore batekin probatzen

2. Ariketa. Liburutegi estatikoak eta dinamikoak

1. Bilatu Interneten “*Creating a shared and static library with the gnu compiler...*” artikulua
2. Sortu liburutegi estatiko eta dinamiko bana **pi_kalkulatu.c** fitxategiarekin
3. Probatu aurreko bi liburutegiak **pi_test.c** programaren bidez (programaren bi bertsio sortuz). Konparatu fitxategi exekutagarrien tamainak
4. Aztertu aurreko bi programen exekuzio denborak UNIXeko *time* tresnaren bidez, 10000000 eta 300000000 balioak erabiliz. Zer ondorio atera dezakezu?

```
[acaf0130@g000008 Lab5]$ cat pi_kalkulatu.h
////////////////////
// pi_kalkulatu.h
//
// libpi liburutegirako
////////////////////

void funtzioa(double *t, double *k, double *l);
```

```
[acaf0130@g000008 Lab5]$ cat pi_kalkulatu.c
////////////////////
// pi_kalkulatu.c
//
// libpi liburutegirako
////////////////////

// Formula:  $Pi = 4 * (1 - 1/3 + 1/5 - 1/7 + 1/9 - \dots)$ 

void funtzioa(double *t, double *k, double *l)
{
    (*t) += (*l)/(*k);
    (*k) += 2.0;
    (*l) *= -1.0;
}
```

```
[acaf0130@g000008 Lab5]$
```

```

[acaf0130@g000008 Lab5]$ cat pi_test.c
////////////////////////////////////
// pi_test.c
//
// Erabilpena:   pi_test   iterazio_kop
////////////////////////////////////

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

#include "pi_kalkulatu.h"

int main(int argc, char *argv[])
{
    double t, k = 3.0, l = -1.0;
    int i, s;

    if (argc < 2) {
        fprintf(stderr, "%s <iterazio_kopurua>\n", argv[0]);
        exit(1);
    }

    t = 1.0;
    for (i = 0, s = atoi(argv[1]); i < s; i++) {
        funtzioa(&t, &k, &l);
    }
    t *= 4;

    printf("Kalkulatutako Pi: %1.16f, math.h fitxategiko Pi: %.16f\n", t, M_PI);
    printf("Diferentzia Absolutua: %.16f\n", fabs(M_PI - t));

    return 0;
}

```

3. Ariketa. Programen kargarako sistema-deiak

1. Probatu **jaurti0.c** `system()` erabiliz
Probatu “**erlojua 30&**” eta “**ps -f**”
aginduak, Nor da erlojuaren gurasoa?
2. Probatu **jaurti0b.c** `execlp()` erabiliz
3. Konparatu bi bertsioak

```
[acaf0130@g000008 Lab5]$ cat jaurti0.c
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>

#define errore(a) {perror(a); exit(1);};
#define BUFSIZE 512

main (int argc, char *argv[])
{
    int n;
    char buf[BUFSIZE];

    write(1, "Jaurti0> ", strlen("Jaurti0> "));
    while ((n = read(0, buf, BUFSIZE)) > 0) {
        buf[n-1] = '\0';
        system(buf);
        write(1, "Jaurti0> ", strlen("Jaurti0> "));
    }
    if (n == -1) errore("read");
    write(1, "\n", 1);
}
```

```
[acaf0130@g000008 Lab5]$ cat jaurti0b.c
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>
#include <unistd.h>

#define errore(a) {perror(a); exit(1);};
#define BUFSIZE 512

main (int argc, char *argv[])
{
    int n;
    char buf[BUFSIZE];

    write(1, "Jaurti0> ", strlen("Jaurti0> "));
    while ((n = read(0, buf, BUFSIZE)) > 0) {
        buf[n-1] = '\0';
        // system(buf);
        execlp(buf, buf, NULL);
        write(1, "Jaurti0> ", strlen("Jaurti0> "));
    }
    if (n == -1) errore("read");
    write(1, "\n", 1);
}
```