

# SISTEMAS OPERATIVOS I

## Ficha 7

Abril de 2006

Nuno Malheiro  
Maria João Viamonte  
Berta Batista  
Luis Lino Ferreira

Sugestões e participações de erros para:  
[ntm@dei.isep.ipp.pt](mailto:ntm@dei.isep.ipp.pt)

## 1 O debug simples de um programa em C

- 1.1 Coloque o código fonte se encontra a seguir, num ficheiro chamado "ola.c". Compile-o utilizando o gcc, com o comando que lhe segue. Execute o gdb usando a linha final e efectue uma execução passo a passo.

```
#include <stdio.h>

void ola (const char * nome)
{
    printf ("Ola, %s!\n", nome);
}

int main (void)
{
    ola ("Mundo");
    return 0;
}
```

```
$ gcc -g ola.c -o ola
```

```
$ gdb ola
```

- 1.2 Coloque o código fonte se encontra a seguir, num ficheiro chamado "mau.c". Encontre os erros do programa seguinte, compilando-o e efectuando o seu debug passo a passo, começando na função fnc. Corrija o programa, recompile-o e execute-o.

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

void fnc(char *n)
{
    char *l;
    l = malloc(strlen(n)*sizeof(char));
    strcpy(l,n);
    printf("Valor de n: %s\n",n);
    free(l);
}

int main(void)
{
    fnc("SOP1");
    fnc(NULL);
    return 0;
}
```

- 1.3 Coloque o código fonte se encontra a seguir, num ficheiro chamado "ciclo.c". Usando o GDB, execute o programa parando-o no início da função ciclo. Estabeleça um ponto de paragem para quando a variável do ciclo for maior do que 7. Continue a execução até essa condição. Nessa altura verifique todas as variáveis e a stack de execução. Continue o programa até ao final.

```

#include <stdio.h>

void ciclo(int *v, int n)
{
    int i =0;
    for(i=0 ;i<n ;i++)
    {
        printf("Valor de i: %i\n",v[i]);
        v[i]=0;
    }
}
int main(void)
{
    int v[10] = { 1, 2, 4, 8, 16, 32, 64, 128, 256, 512 };
    ciclo(v, 10);
    return 0;
}

```

## 2 Exemplo mais avançado de debug de um programa em C

- 2.1 Coloque o código fonte se encontra a seguir, num ficheiro chamado "lista.c". Usando o GDB, execute o programa com os argumentos "A1" e "A2", parando-o no início da função main. Tente executar passo a passo a função myfunc. Na execução dessa função, verifique a stack e o vector udata.

```

#include <glib.h>
#include <stdio.h>

void myfunc(char *i, char **udata)
{
    printf("%s ; %s : %s\n", udata[1], udata[2], i);
}

int main(int argc, char *argv[])
{
    GList *list = NULL;

    list = g_list_append(list, "Elemento 1");
    list = g_list_append(list, "Elemento 2");

    g_list_foreach(list, (GFunc) myfunc, argv);
    return 0;
}

```

- 2.2 Usando o programa anterior, execute-o com os argumentos "A1" e "A2". Dentro da execução altere os argumentos para "B1" e "B2" na primeira execução da função "myfunc" e para "C1" e "C2" na segunda execução de "myfunc"