

**Instituto Superior de Engenharia de Lisboa**  
Licenciatura em Engenharia Informática e de Computadores

# **Sistemas Operativos**

Processo

# Execução de Programas em Simultâneo

Cada programa, para se executar, necessita apenas de parte dos recursos:

- Ocupa apenas parte da memória
- Não precisa de 100% do tempo de CPU (há, pelo menos, esperas por I/O)

Assim, é possível lançar várias execuções em simultâneo:

- Atribuindo parte da memória física a cada programa em execução
- Distribuindo o tempo de processamento do CPU pelas várias execuções

Para executar um programa, o sistema cria um ambiente de execução específico, designado por **processo**.

# Suporte para Múltiplos Utilizadores

Em ambientes empresariais, em que o investimento num único computador era, só por si, uma despesa avultada, pretendia-se:

- Fazer chegar a capacidade computacional ao máximo de pessoas
- Maximizar a utilização dos recursos computacionais

Nesse sentido, os sistemas operativos permitiam conectar múltiplos terminais ao mesmo computador.

- Cada terminal tem apenas capacidade de input/output de dados (ex.: teclado e ecrã), estando a capacidade computacional centralizada.
- Cada utilizador usa um terminal para executar os programas de que precisa

# Computador e Terminais



Computador: DEC PDP-7



Teletype Model 33



Terminal: DEC VT-52

# Compilação e ligação com localização

```
const int a = 3, b = 4;
int c = -1;

double f1(double x, double y) {
    return sqrt(x*x + y*y);
}

void show(int x, int y, int z) {
    printf("x: %d; y: %d; z: %d\n", x, y, z);
}

int main() {
    printf("  a: %p\n", &a);
    printf("  b: %p\n", &b);
    printf("  c: %p\n", &c);
    printf(" f1: %p\n", f1);
    printf("show: %p\n", show);
    c = (int)f1(a, b);
    show(a, b, c);
    return 0;
}
```



```
$ gcc -o prog prog.c -lm
$ nm prog
00000000004007d4 R a
00000000004007d8 R b
...
0000000000601040 D c
...
0000000000400626 T f1
0000000000400669 T show
...
```



```
$ ./prog
a: 0x4007d4
b: 0x4007d8
c: 0x601040
f1: 0x400626
show: 0x400669
x: 3; y: 4; z: 5
```

Endereços  
decididos  
durante a  
produção do  
executável.

# Distribuição e Controlo de Acesso a Recursos

- CPU(s) distribuído(s) ao longo do tempo pelos vários fios de execução
- Um fio de execução não pode monopolizar o uso de um CPU
- Memória física deve ser distribuída pelos vários programas em execução
- Um programa em execução não pode aceder livremente à memória de outros
- O acesso a dispositivos de I/O é mediado pelo sistema
- É necessário controlar permissões e acessos em exclusivo

# Processo como Ambiente Virtual de Execução

- **Espaço de endereçamento** virtualiza acesso à memória física
- **Threads** (incluindo a principal) virtualizam acesso ao(s) CPU(s)
- **Descritores de ficheiros** virtualizam acesso a dispositivos de I/O

# Processo

