Controlo de Processos

José C. Cunha, DI/FCT/UNL

Ambiente de execução dos programas:

- instruções da máquina hardware
- chamadas a funções de biblioteca
- chamadas ao sistema de operação

+ Suporte implícito da máquina virtual dedicada a cada processo:

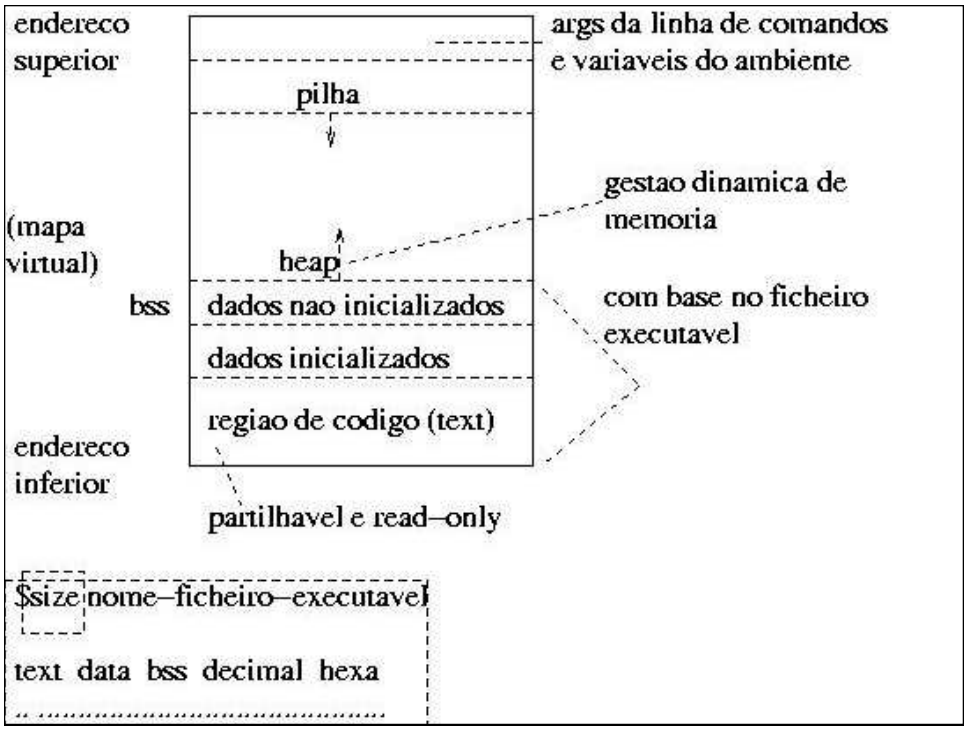
--- Processador Virtual

--- Memória Virtual

--- Canais Virtuais de Comunicação

Ambiente de um processo Unix

- Mapa de memória do processo
- Inicialização -> main()
- Execução e terminação do processo



main()

int main(int argc, char *argv[ ]);

-- número de argumentos

-- vector de apontadores para os argumentos

```
{ int i;
  for (i = 0; i<argc; i++)
     printf("argv[%d]: %s\n", i, argv[i]);
}
```

Uma chamada ao sistema exec() pode passar argumentos a um novo programa.

Activar um programa C:

-- uma função de inicialização (no “start address” especificado pelo compilador)

é activada, obtém os argumentos/vars ambiente

para o mapa do processo,

e lança a função main().

Terminação de processos

- Normal:

-- retorno de main()

-- chamada de exit: acções de ‘fecho’, …

-- chamada de _exit(): retorno imediato ao SO

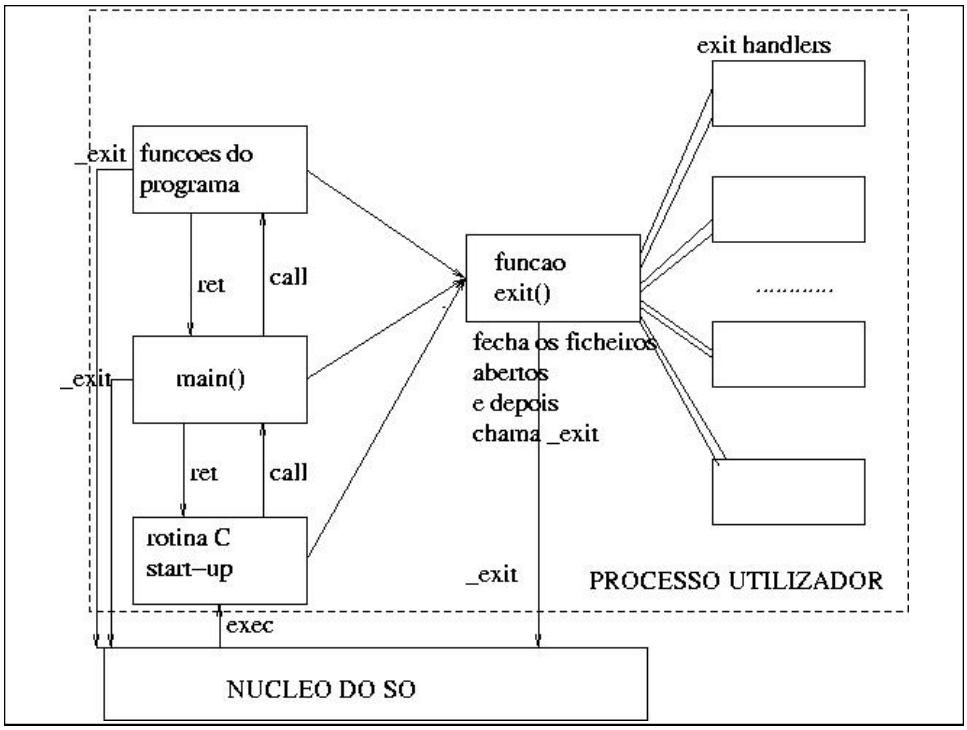
--podem devolver o ‘exit status’ do processo

- Anormal:

-- chamada de abort()

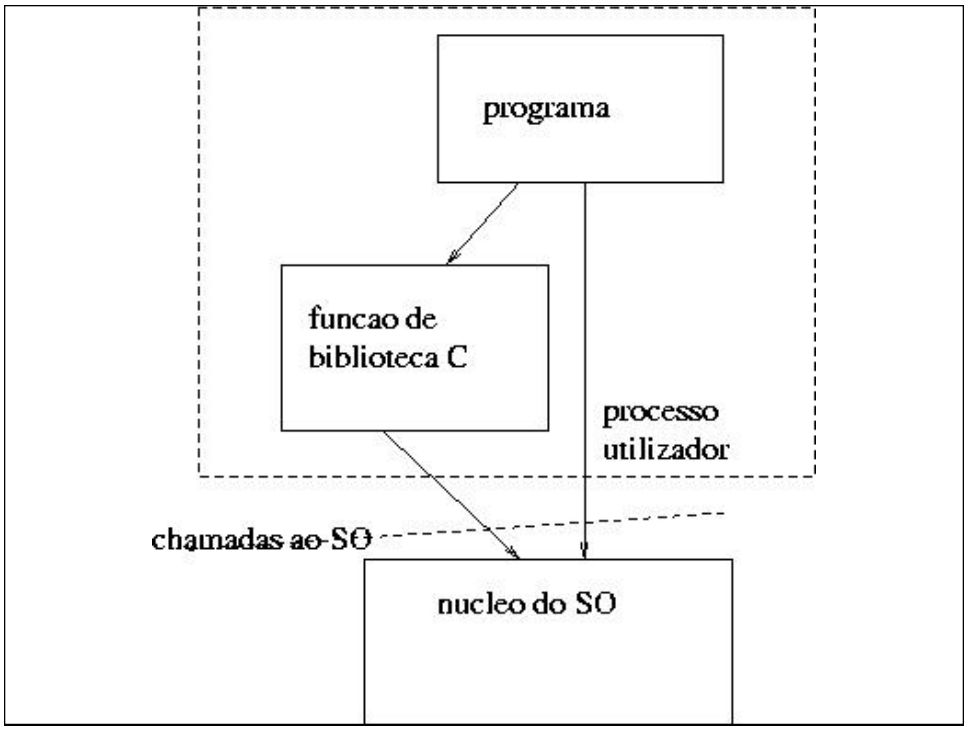
-- ocorrência de um sinal

(hardware ou software)



Chamadas a biblioteca e chamadas ao sistema

- As chamadas ao sistema têm funções correspondentes na bib. std C
- Muitas funções de biblioteca não chamam o sistema: operam em MODO UTILIZADOR!
- As chamadas ao sistema operam em MODO SUPERVISOR!
- Exemplo:

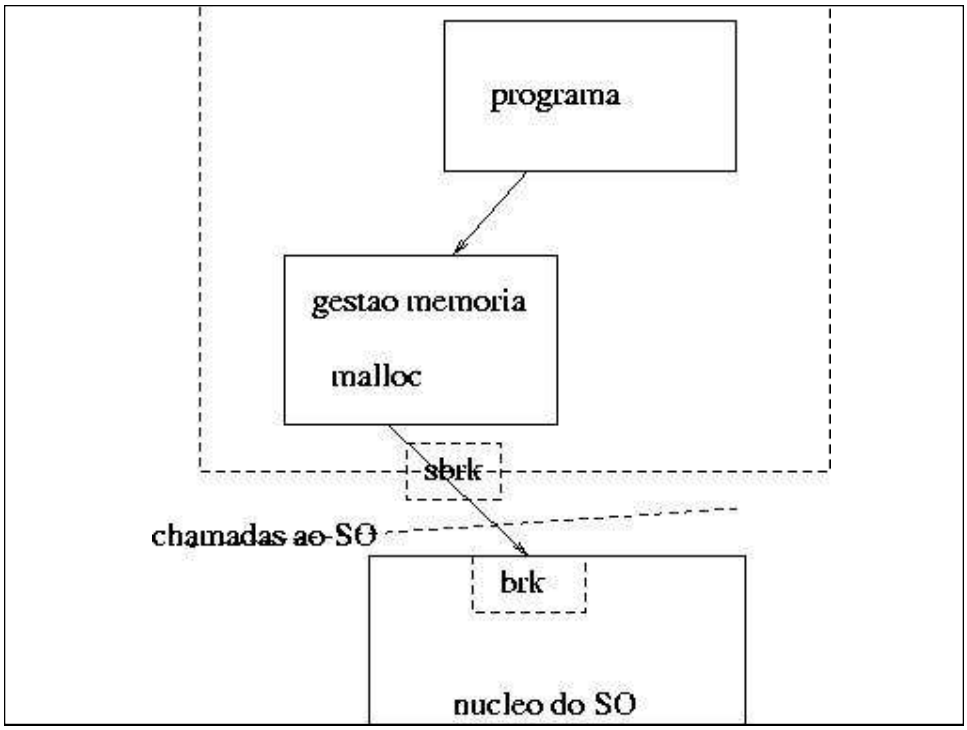


Gestão de Memória

1. Malloc: reserva um certo n. de bytes
2. Calloc: reserva espaço para um certo n. de objectos e inicializa
3. Realloc: aumenta/reduz o tamanho de uma área antes reservada
4. Free: liberta espaço apontado por um argumento

-- implementadas sobre chamada ao SO

brk, que ajusta o tamanho do HEAP


programa
gestao memoria
malloc
sbrk
chamadas ao SO
brk
nucleo do SO