

Sistemas Operativos

Apontadores e debugging

Grupo de Sistemas Distribuídos
Universidade do Minho

1 Objectivos

Exercitar o uso de apontadores em C, implementando uma lista ligada. Familiarização com ferramentas de debugging.

2 Ferramentas a utilizar

GNU debugger (gdb) para execução passo a passo, inspecção de variáveis e da causa de *segmentation faults*.

Valgrind para validação da utilização de apontadores e primitivas de gestão de memória.

3 Exercícios propostos

Considere uma aplicação que faz a gestão de intervalos de lugares marcados durante uma época num recinto desportivo. Admita que os lugares estão numerados entre 0 e 65535. Use uma lista ligada em que cada elemento contém um intervalo de lugares disponíveis. Por exemplo, quando não há lugares reservados, deve conter um elemento com $[0, +65535]$. Quando os lugares 100 a 109 (inclusivé) estão ocupados, a lista deve conter dois elementos com $[0, +100]$ e $[110, +65425]$.

3.1 Implemente em C uma biblioteca tão simples quanto possível que cumpra os objectivos com as seguintes funções:

```
struct intv* reserva(struct intv* livres, int n, int* reservado);  
struct intv* liberta(struct intv* livres, int lugar, int n);
```

A função `reserva` deve colocar o número do primeiro lugar reservado no pântmetro `reservado`. Caso não haja lugares, deve colocar -1 .

3.2 Implemente uma nova versão da biblioteca. Desta vez deverá ser utilizada dupla indirectão, em vez de ser devolvido o novo início da lista.

```
int reserva(struct intv** livres, int n);  
void liberta(struct intv** livres, int lugar, int n);
```

A função `reserva` deve devolver o número do primeiro lugar reservado, ou -1 caso não haja lugares disponíveis.

4 Exercícios adicionais

4.1 Assegure que a sua implementação é capaz de reservar uma sequência de lugares livres, mesmo quando previamente partes dessa sequência estiveram ocupados por diferentes donos.

4.2 Assegure que uma sequência de lugares pedida é reservada da sequência livre que tem menos lugares excedentários.