

Ponteiros e vetores


Ponteiros e vetores

- A relação que existe entre ponteiros e vetores é forte.
- O compilador entende vetores como ponteiros.

Ponteiros e vetores

Endereço Conteúdo Nome

0x1000		
0x1004		
0x1008		
0x1012		
0x1016		
0x1020		
0x1024		
0x1028		
0x1032		
0x1036		
0x1040		
0x1044		
0x1048		
0x1052		
0x1056		
0x1060		



```
int main(void) {  
  
    int vetor[4] = {5,2,7,8};  
  
    printf("%p\n", vetor);  
    printf("%p\n", &vetor[0]);  
    printf("%i\n", *vetor);  
    printf("%i\n", *(vetor + 1));  
  
    int* p = vetor;  
  
    p = p + 2;  
  
    printf("%p\n", p);  
    printf("%i\n", *p);  
  
    p = p - 1;  
    *p = 20;  
  
    printf("%p\n", p);  
    printf("%i\n", p[0]);  
    printf("%i\n", p[1]);  
    printf("%i\n", p[2]);  
    printf("%i\n", p[3]);  
  
    return 0;  
}
```

Ponteiros e vetores

	Endereço	Conteúdo	Nome
vetor →	0x1000	5	vetor[0]
	0x1004	2	vetor[1]
	0x1008	7	vetor[2]
	0x1012	8	vetor[3]
	0x1016		
	0x1020		
	0x1024		
	0x1028		
	0x1032		
	0x1036		
	0x1040		
	0x1044		
	0x1048		
	0x1052		
	0x1056		
	0x1060		



```
int main(void) {  
  
    int vetor[4] = {5,2,7,8};  
  
    printf("%p\n", vetor);  
    printf("%p\n", &vetor[0]);  
    printf("%i\n", *vetor);  
    printf("%i\n", *(vetor + 1));  
  
    int* p = vetor;  
  
    p = p + 2;  
  
    printf("%p\n", p);  
    printf("%i\n", *p);  
  
    p = p - 1;  
    *p = 20;  
  
    printf("%p\n", p);  
    printf("%i\n", p[0]);  
    printf("%i\n", p[1]);  
    printf("%i\n", p[2]);  
    printf("%i\n", p[3]);  
  
    return 0;  
}
```

Ponteiros e vetores

	Endereço	Conteúdo	Nome
vetor →	0x1000	5	vetor[0]
	0x1004	2	vetor[1]
	0x1008	7	vetor[2]
	0x1012	8	vetor[3]
	0x1016		
	0x1020		
	0x1024		
	0x1028		
	0x1032		
	0x1036		
	0x1040		
	0x1044		
	0x1048		
	0x1052		
	0x1056		
	0x1060		



```
int main(void) {  
  
    int vetor[4] = {5,2,7,8};  
  
    printf("%p\n", vetor);  
    printf("%p\n", &vetor[0]);  
    printf("%i\n", *vetor);  
    printf("%i\n", *(vetor + 1));  
  
    int* p = vetor;  
  
    p = p + 2;  
  
    printf("%p\n", p);  
    printf("%i\n", *p);  
  
    p = p - 1;  
    *p = 20;  
  
    printf("%p\n", p);  
    printf("%i\n", p[0]);  
    printf("%i\n", p[1]);  
    printf("%i\n", p[2]);  
    printf("%i\n", p[3]);  
  
    return 0;  
}
```

Ponteiros e vetores

	Endereço	Conteúdo	Nome
vetor →	0x1000	5	vetor[0]
	0x1004	2	vetor[1]
	0x1008	7	vetor[2]
	0x1012	8	vetor[3]
	0x1016		
	0x1020		
	0x1024		
	0x1028		
	0x1032		
	0x1036		
	0x1040		
	0x1044		
	0x1048		
	0x1052		
	0x1056		
	0x1060		



```
int main(void) {  
  
    int vetor[4] = {5,2,7,8};  
  
    printf("%p\n", vetor);  
    printf("%p\n", &vetor[0]);  
    printf("%i\n", *vetor);  
    printf("%i\n", *(vetor + 1));  
  
    int* p = vetor;  
  
    p = p + 2;  
  
    printf("%p\n", p);  
    printf("%i\n", *p);  
  
    p = p - 1;  
    *p = 20;  
  
    printf("%p\n", p);  
    printf("%i\n", p[0]);  
    printf("%i\n", p[1]);  
    printf("%i\n", p[2]);  
    printf("%i\n", p[3]);  
  
    return 0;  
}
```

Ponteiros e vetores

	Endereço	Conteúdo	Nome
vetor →	0x1000	5	vetor[0]
	0x1004	2	vetor[1]
	0x1008	7	vetor[2]
	0x1012	8	vetor[3]
	0x1016		
	0x1020		
	0x1024		
	0x1028	0x1000	p
	0x1032		
	0x1036		
	0x1040		
	0x1044		
	0x1048		
	0x1052		
	0x1056		
	0x1060		

```
int main(void) {  
  
    int vetor[4] = {5,2,7,8};  
  
    printf("%p\n", vetor);  
    printf("%p\n", &vetor[0]);  
    printf("%i\n", *vetor);  
    printf("%i\n", *(vetor + 1));  
  
    int* p = vetor;  
  
    p = p + 2;  
  
    printf("%p\n", p);  
    printf("%i\n", *p);  
  
    p = p - 1;  
    *p = 20;  
  
    printf("%p\n", p);  
    printf("%i\n", p[0]);  
    printf("%i\n", p[1]);  
    printf("%i\n", p[2]);  
    printf("%i\n", p[3]);  
  
    return 0;  
}
```

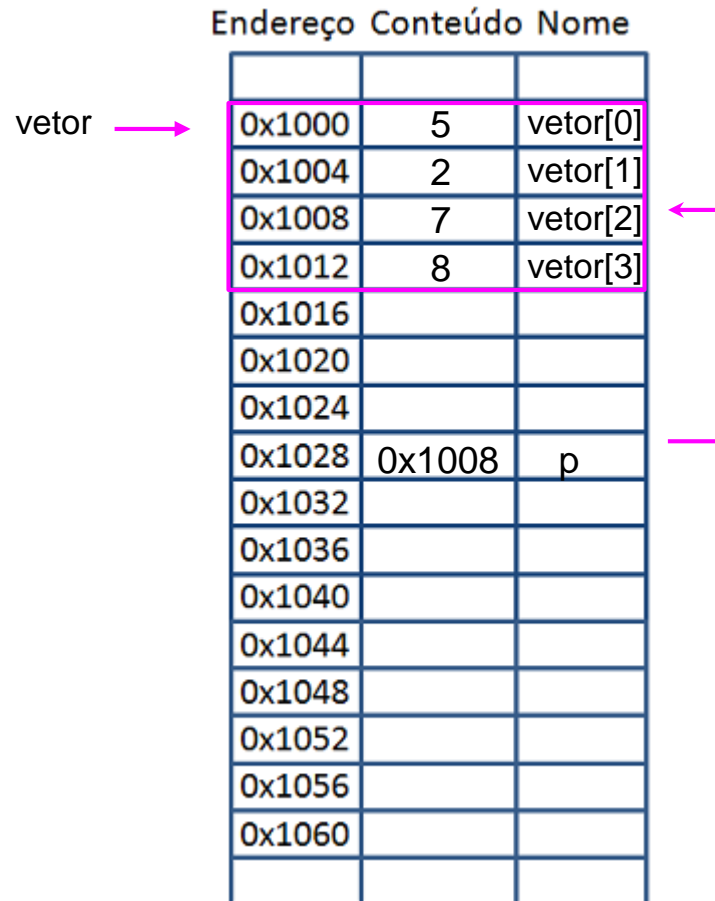
Ponteiros e vetores

	Endereço	Conteúdo	Nome
vetor →	0x1000	5	vetor[0]
	0x1004	2	vetor[1]
	0x1008	7	vetor[2]
	0x1012	8	vetor[3]
	0x1016		
	0x1020		
	0x1024		
	0x1028	0x1000	p
	0x1032		
	0x1036		
	0x1040		
	0x1044		
	0x1048		
	0x1052		
	0x1056		
	0x1060		

```
int main(void) {  
  
    int vetor[4] = {5,2,7,8};  
  
    printf("%p\n", vetor);  
    printf("%p\n", &vetor[0]);  
    printf("%i\n", *vetor);  
    printf("%i\n", *(vetor + 1));  
  
    int* p = vetor;  
  
    p = p + 2;  
  
    printf("%p\n", p);  
    printf("%i\n", *p);  
  
    p = p - 1;  
    *p = 20;  
  
    printf("%p\n", p);  
    printf("%i\n", p[0]);  
    printf("%i\n", p[1]);  
    printf("%i\n", p[2]);  
    printf("%i\n", p[3]);  
  
    return 0;  
}
```


Ponteiros e vetores

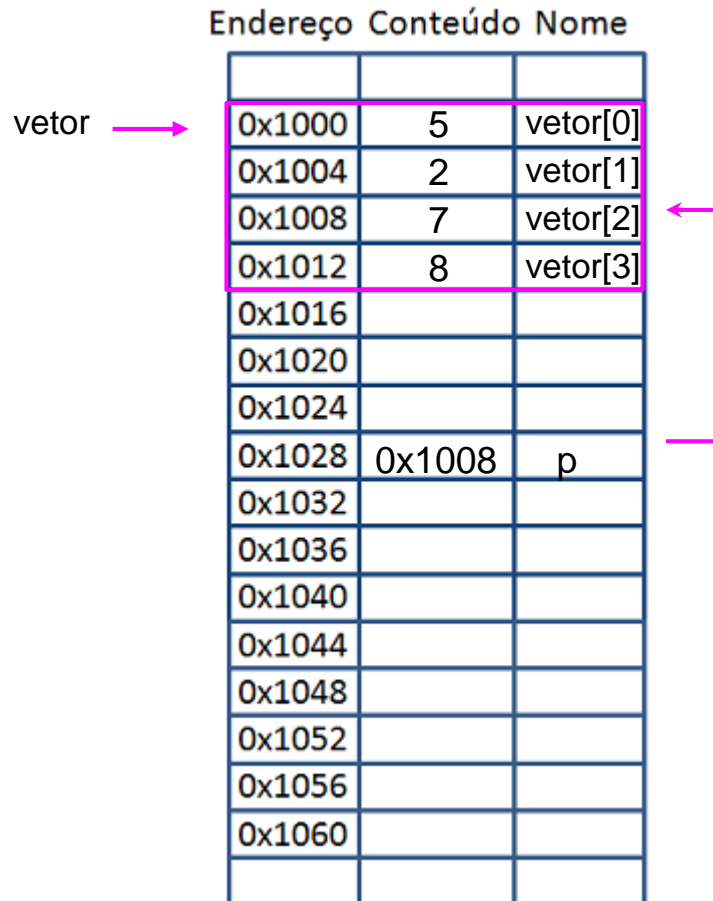
	Endereço	Conteúdo	Nome
vetor →	0x1000	5	vetor[0]
	0x1004	2	vetor[1]
	0x1008	7	vetor[2]
	0x1012	8	vetor[3]
	0x1016		
	0x1020		
	0x1024		
	0x1028	0x1008	p
	0x1032		
	0x1036		
	0x1040		
	0x1044		
	0x1048		
	0x1052		
	0x1056		
	0x1060		



```
int main(void) {  
  
    int vetor[4] = {5,2,7,8};  
  
    printf("%p\n", vetor);  
    printf("%p\n", &vetor[0]);  
    printf("%i\n", *vetor);  
    printf("%i\n", *(vetor + 1));  
  
    int* p = vetor;  
  
    p = p + 2;  
  
    printf("%p\n", p);  
    printf("%i\n", *p);  
  
    p = p - 1;  
    *p = 20;  
  
    printf("%p\n", p);  
    printf("%i\n", p[0]);  
    printf("%i\n", p[1]);  
    printf("%i\n", p[2]);  
    printf("%i\n", p[3]);  
  
    return 0;  
}
```

Ponteiros e vetores

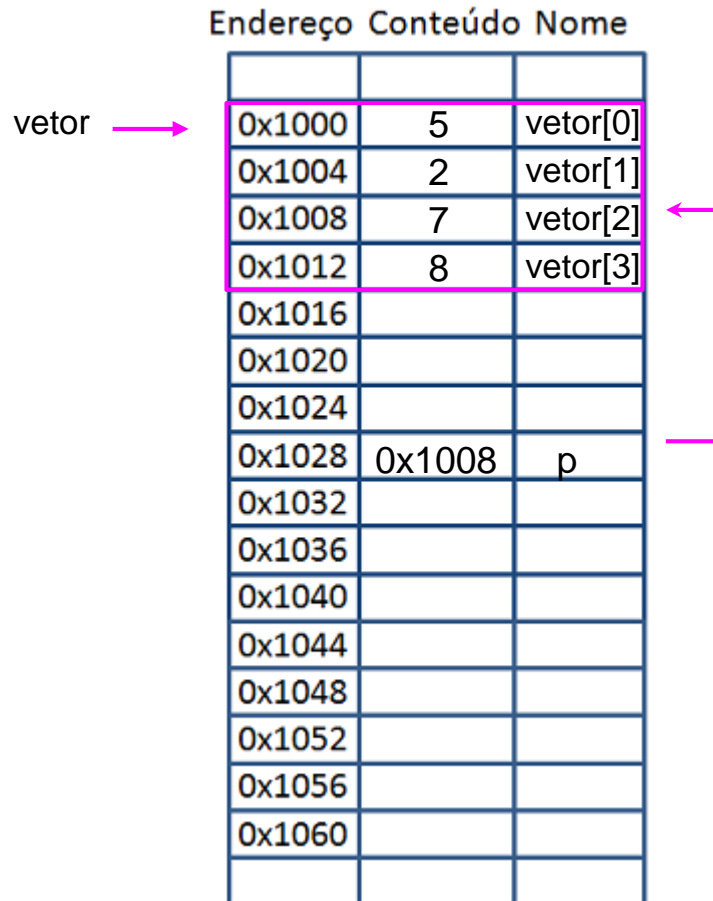
	Endereço	Conteúdo	Nome
vetor →	0x1000	5	vetor[0]
	0x1004	2	vetor[1]
	0x1008	7	vetor[2]
	0x1012	8	vetor[3]
	0x1016		
	0x1020		
	0x1024		
	0x1028	0x1008	p
	0x1032		
	0x1036		
	0x1040		
	0x1044		
	0x1048		
	0x1052		
	0x1056		
	0x1060		



```
int main(void) {  
  
    int vetor[4] = {5,2,7,8};  
  
    printf("%p\n", vetor);  
    printf("%p\n", &vetor[0]);  
    printf("%i\n", *vetor);  
    printf("%i\n", *(vetor + 1));  
  
    int* p = vetor;  
  
    p = p + 2;  
  
    printf("%p\n", p);  
    printf("%i\n", *p);  
  
    p = p - 1;  
    *p = 20;  
  
    printf("%p\n", p);  
    printf("%i\n", p[0]);  
    printf("%i\n", p[1]);  
    printf("%i\n", p[2]);  
    printf("%i\n", p[3]);  
  
    return 0;  
}
```

Ponteiros e vetores

	Endereço	Conteúdo	Nome
vetor →	0x1000	5	vetor[0]
	0x1004	2	vetor[1]
	0x1008	7	vetor[2]
	0x1012	8	vetor[3]
	0x1016		
	0x1020		
	0x1024		
	0x1028	0x1008	p
	0x1032		
	0x1036		
	0x1040		
	0x1044		
	0x1048		
	0x1052		
	0x1056		
	0x1060		



```
int main(void) {  
  
    int vetor[4] = {5,2,7,8};  
  
    printf("%p\n", vetor);  
    printf("%p\n", &vetor[0]);  
    printf("%i\n", *vetor);  
    printf("%i\n", *(vetor + 1));  
  
    int* p = vetor;  
  
    p = p + 2;  
  
    printf("%p\n", p);  
    printf("%i\n", *p);  
  
    p = p - 1;  
    *p = 20;  
  
    printf("%p\n", p);  
    printf("%i\n", p[0]);  
    printf("%i\n", p[1]);  
    printf("%i\n", p[2]);  
    printf("%i\n", p[3]);  
  
    return 0;  
}
```

Ponteiros e vetores

	Endereço	Conteúdo	Nome
vetor →	0x1000	5	vetor[0]
	0x1004	2	vetor[1]
	0x1008	7	vetor[2]
	0x1012	8	vetor[3]
	0x1016		
	0x1020		
	0x1024		
	0x1028	0x1004	p
	0x1032		
	0x1036		
	0x1040		
	0x1044		
	0x1048		
	0x1052		
	0x1056		
	0x1060		

```
int main(void) {  
  
    int vetor[4] = {5,2,7,8};  
  
    printf("%p\n", vetor);  
    printf("%p\n", &vetor[0]);  
    printf("%i\n", *vetor);  
    printf("%i\n", *(vetor + 1));  
  
    int* p = vetor;  
  
    p = p + 2;  
  
    printf("%p\n", p);  
    printf("%i\n", *p);  
  
    p = p - 1;  
    *p = 20;  
  
    printf("%p\n", p);  
    printf("%i\n", p[0]);  
    printf("%i\n", p[1]);  
    printf("%i\n", p[2]);  
    printf("%i\n", p[3]);  
  
    return 0;  
}
```

Ponteiros e vetores

	Endereço	Conteúdo	Nome
vetor →	0x1000	5	vetor[0]
	0x1004	2	vetor[1]
	0x1008	7	vetor[2]
	0x1012	8	vetor[3]
	0x1016		
	0x1020		
	0x1024		
	0x1028	0x1004	p
	0x1032		
	0x1036		
	0x1040		
	0x1044		
	0x1048		
	0x1052		
	0x1056		
	0x1060		

```
int main(void) {  
  
    int vetor[4] = {5,2,7,8};  
  
    printf("%p\n", vetor);  
    printf("%p\n", &vetor[0]);  
    printf("%i\n", *vetor);  
    printf("%i\n", *(vetor + 1));  
  
    int* p = vetor;  
  
    p = p + 2;  
  
    printf("%p\n", p);  
    printf("%i\n", *p);  
  
    p = p - 1;  
    *p = 20;  
  
    printf("%p\n", p);  
    printf("%i\n", p[0]);  
    printf("%i\n", p[1]);  
    printf("%i\n", p[2]);  
    printf("%i\n", p[3]);  
  
    return 0;  
}
```

Ponteiros e vetores

	Endereço	Conteúdo	Nome
vetor →	0x1000	5	vetor[0]
	0x1004	20	vetor[1]
	0x1008	7	vetor[2]
	0x1012	8	vetor[3]
	0x1016		
	0x1020		
	0x1024		
	0x1028	0x1004	p
	0x1032		
	0x1036		
	0x1040		
	0x1044		
	0x1048		
	0x1052		
	0x1056		
	0x1060		

```
int main(void) {  
  
    int vetor[4] = {5,2,7,8};  
  
    printf("%p\n", vetor);  
    printf("%p\n", &vetor[0]);  
    printf("%i\n", *vetor);  
    printf("%i\n", *(vetor + 1));  
  
    int* p = vetor;  
  
    p = p + 2;  
  
    printf("%p\n", p);  
    printf("%i\n", *p);  
  
    p = p - 1;  
    *p = 20;  
  
    printf("%p\n", p);  
    printf("%i\n", p[0]);  
    printf("%i\n", p[1]);  
    printf("%i\n", p[2]);  
    printf("%i\n", p[3]);  
  
    return 0;  
}
```

Ponteiros e vetores

	Endereço	Conteúdo	Nome
vetor →	0x1000	5	vetor[0]
	0x1004	20	vetor[1]
	0x1008	7	vetor[2]
	0x1012	8	vetor[3]
	0x1016		
	0x1020		
	0x1024		
	0x1028	0x1004	p
	0x1032		
	0x1036		
	0x1040		
	0x1044		
	0x1048		
	0x1052		
	0x1056		
	0x1060		

```
int main(void) {  
  
    int vetor[4] = {5,2,7,8};  
  
    printf("%p\n", vetor);  
    printf("%p\n", &vetor[0]);  
    printf("%i\n", *vetor);  
    printf("%i\n", *(vetor + 1));  
  
    int* p = vetor;  
  
    p = p + 2;  
  
    printf("%p\n", p);  
    printf("%i\n", *p);  
  
    p = p - 1;  
    *p = 20;  
  
    printf("%p\n", p);  
    printf("%i\n", p[0]);  
    printf("%i\n", p[1]);  
    printf("%i\n", p[2]);  
    printf("%i\n", p[3]);  
  
    return 0;  
}
```