

12-03-2010

LEI-T-11

Pedro Miguel Carvalho

49386

Stack.h
Stack.c
Módulo

stack.o

gcc -c stack.c

prog3.c

```
#include <stdio.h>
#include "stack.h" → o ficheiro tem de estar
na directoria corrente
```

```
int main ()
```

```
{
```

```
Stack s1, s2 = {{1, 2, 3, 4, 5}, 5};
```

```
Stack ls [10] = {{7, 8}, {9, 10}, 2};
```

```
s1 = push (s1, 13);
```

```
s1 = push (push (push (s1, 27), 19), 21);
```

```
ls [1] = pop (ls [0]);
```

```
s1 = push (pop (ls [0]), 7);
```

```
}
```

array de stack de dimensão 10 com as 2 primeiras posições preenchidas

Para criar o executável:

gcc -c prog3.c → prog3.o

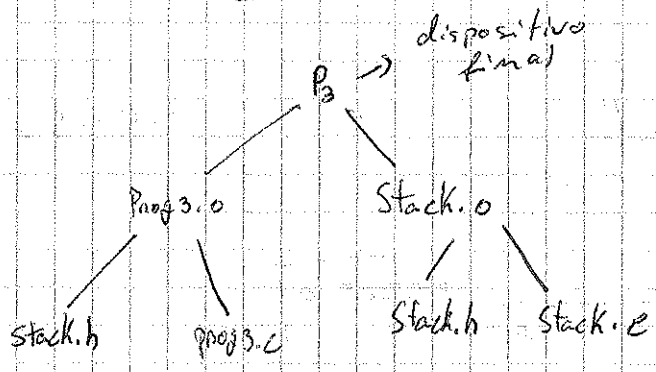
stack.h
stack.c
stack.o
prog3.c

⇒

gcc -o p3 prog3.o stack.c

Make

Albero di dipendenze:



Makefile

temp di sistema
nella colonna ←

```

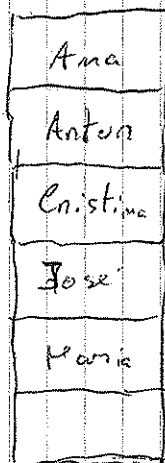
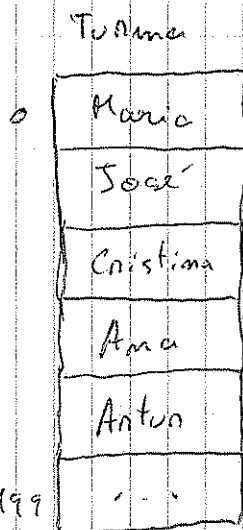
p3: prog3.o stack.o
<TAB> gcc -o p3 prog3.o stack.o
prog3.o: stack.h prog3.c
gcc -c prog3.c
stack.o: stack.h stack.c
gcc -c stack.c
  
```

\$ make

```
typedef struct s Alunos
{
    char nome [60];
    char numero [8];
    char curso [40];
} Aluno; Turma [200];
```

~~typedef~~

```
int main ( )
{
    Turma t1;
    Turma t2 = { {"Maria", "2222", "RI"}, {"Jose", "4258", "EC"} };
}
```



Uma maneira de fazer a pesquisa com eficiencia e comparar com o campo que esta ao meio, eliminando assim metade das comparações.

~~...~~

Neste caso o número médio de comparações para encontrar um aluno é 100.

Uma solução é ordenar o array.

$\frac{n}{2}$

