

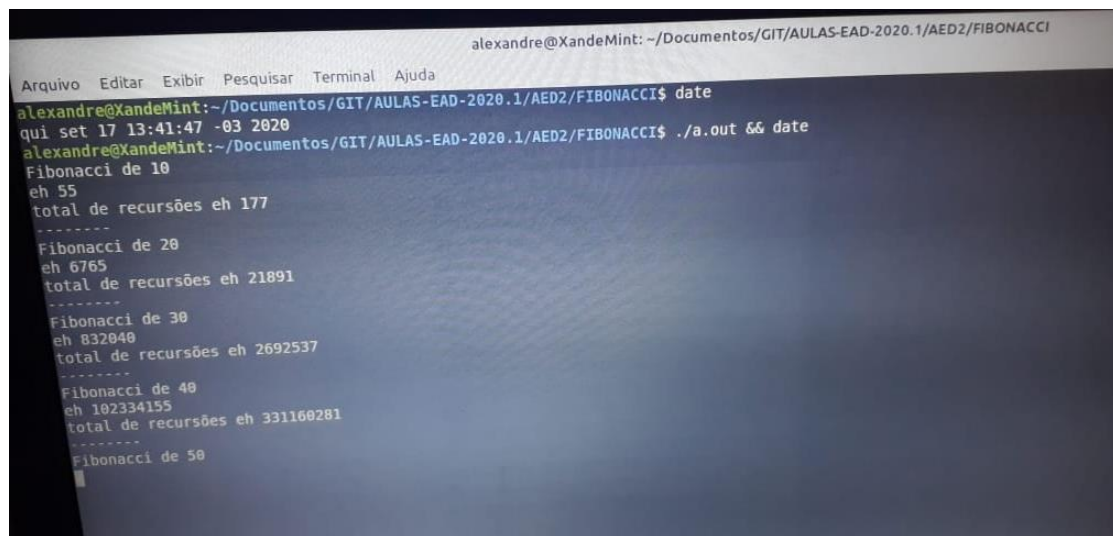
Relatório Fibonacci Recursivo

Inicialmente eu criei o código com um for que itera de 10 em 10, indo de 10 a 100.

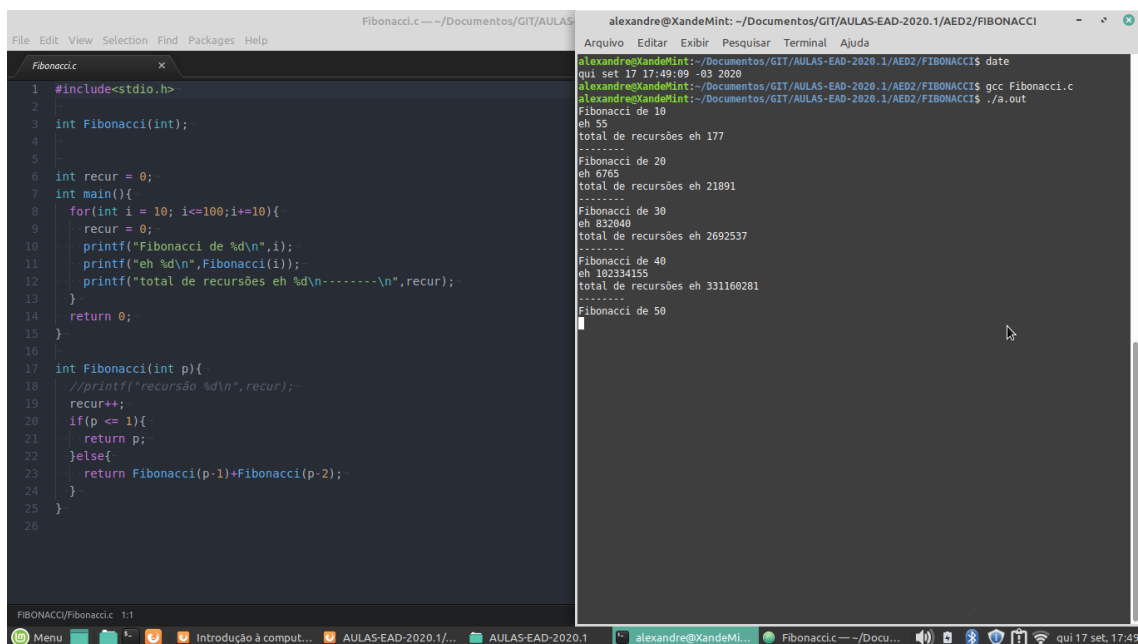
Mas ao executar o código notei que o programa estava demorando demais para terminar sua execução.

Ele parava na execução do Fibonacci de 50, e após alguns segundos iniciava o Fibonacci de 60.

Passaram-se mais de 1 hora de execução e o programa ainda não havia calculado o Fibonacci de 60.



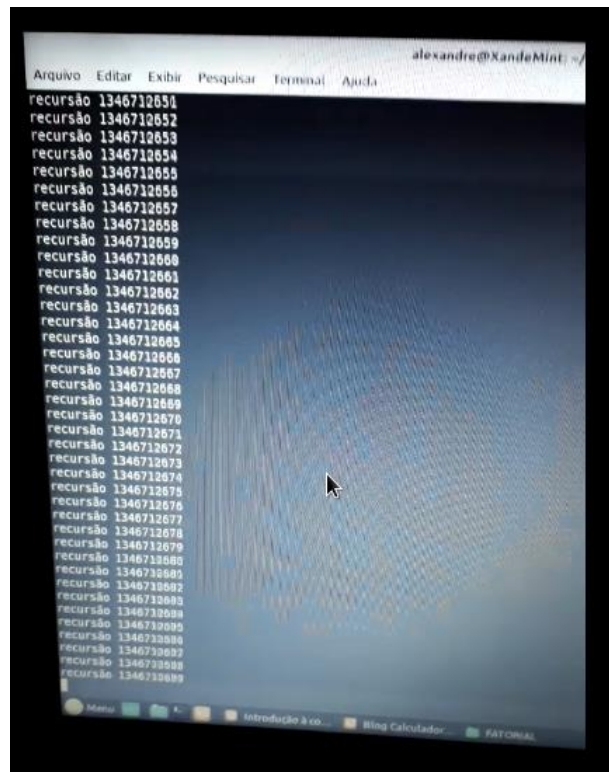
```
alexandre@XandeMint: ~/Documentos/GIT/AULAS-EAD-2020.1/AED2/FIBONACCI
Arquivo Editar Exibir Pesquisar Terminal Ajuda
alexandre@XandeMint:~/Documentos/GIT/AULAS-EAD-2020.1/AED2/FIBONACCI$ date
qui set 17 13:41:47 -03 2020
alexandre@XandeMint:~/Documentos/GIT/AULAS-EAD-2020.1/AED2/FIBONACCI$ ./a.out && date
Fibonacci de 10
eh 55
total de recursões eh 177
-----
Fibonacci de 20
eh 6765
total de recursões eh 21891
-----
Fibonacci de 30
eh 832040
total de recursões eh 2692537
-----
Fibonacci de 40
eh 102334155
total de recursões eh 331160281
-----
Fibonacci de 50
```



```
Fibonacci.c — ~/Documentos/GIT/AULAS-EAD-2020.1/AED2/FIBONACCI
File Edit View Selection Find Packages Help
1 #include<stdio.h>
2
3 int Fibonacci(int);
4
5
6 int recur = 0;
7 int main(){
8     for(int i = 10; i<=100;i+=10){
9         recur = 0;
10        printf("Fibonacci de %d\n",i);
11        printf("eh %d\n",Fibonacci(i));
12        printf("total de recursões eh %d\n-----\n",recur);
13    }
14    return 0;
15 }
16
17 int Fibonacci(int p){
18     //printf("recursão %d\n",recur);
19     recur++;
20     if(p <= 1){
21         return p;
22     }else{
23         return Fibonacci(p-1)+Fibonacci(p-2);
24     }
25 }
26
```

Querendo entender o motivo da demora, alterei o código para imprimir na tela a cada nova recursão iniciada e assim averiguar se o programa estava travando.

Novamente após 1 hora de execução pude notar que a quantidade de recursões já estava acima de 1346763324 e continuava sem parar a entrar em outras recursões.



Analisando melhor, pude observar que o Fibonacci de 10 necessitava de 55 recursões e que o Fibonacci de 20 necessitava de 6765 recursões e o de 30 precisava de 832040, logo percebi que o Fibonacci de 60 seria um número muito maior, e após constatar que após 1 hora de execução o valor ainda não tinha sido encontrado, encerrei os testes pois possivelmente este programa precisaria de no mínimo alguns dias para achar o valor de 60.

Cheguei a conclusão que este programa seria incapaz de ser executado pois possivelmente ele levaria mais anos que eu para terminar, visto que o mesmo ainda iria calcular o Fibonacci de 70, 80, 90 e 100.

```

1 #include<stdio.h>
2
3 int Fibonacci(int);
4
5
6 int recur = 0;
7 int main(){
8     for(int i = 10; i<=100;i+=10){
9         recur = 0;
10        printf("Fibonacci de %d\n",i);
11        printf("eh %d\n",Fibonacci(i));
12        printf("total de recursões eh %d\n-----\n",recur);
13    }
14    return 0;
15 }
16
17 int Fibonacci(int p){
18     //printf("recursão %d\n",recur);
19     recur++;
20     if(p <= 1){
21         return p;
22     }else{
23         return Fibonacci(p-1)+Fibonacci(p-2);
24     }
25 }
26

```

```

alexandre@XandeMint: ~/Documentos/GIT/AULAS-EAD-2020.1/AED2/FIBONACCI$ date
qui set 17 17:49:09 -03 2020
alexandre@XandeMint:~/Documentos/GIT/AULAS-EAD-2020.1/AED2/FIBONACCI$ gcc Fibonacci.c
alexandre@XandeMint:~/Documentos/GIT/AULAS-EAD-2020.1/AED2/FIBONACCI$ ./a.out
Fibonacci de 10
eh 55
total de recursões eh 177
-----
Fibonacci de 20
eh 6765
total de recursões eh 21891
-----
Fibonacci de 30
eh 832040
total de recursões eh 2692537
-----
Fibonacci de 40
eh 102334155
total de recursões eh 331160281
-----
Fibonacci de 50
eh -298632863
total de recursões eh 2075316483
-----
Fibonacci de 60

```

Oque torna seu uso inviável.