

- [Questões Básicas](#)
- [Processos](#)
- [Gerando uma árvore de processos](#)
- [Comunicando por FIFO - atividade de 4/4](#)

Questões Básicas

1. Enumere e explique resumidamente as principais responsabilidades de um sistema operacional.
2. Qual é a diferença entre modo usuário e modo kernel? Por que esta distinção é feita?
3. Descreva como são implementados os seguintes eventos: (i) chamada de sistema, (ii) tratamento de interrupção e (iii) tratamento de sinais. Quais são as semelhanças e diferenças entre estes eventos?
4. Qual é a diferença entre um sistema monolítico e um baseado em camadas?

Processos

1. Descreva o funcionamento da função `fork()`. Após o `fork`, como os processos pai e filho podem se comunicar/sincronizar?
2. O que é um processo Zumbi? Escreva um programa que crie processos zumbis.
3. Familiarize-se com as chamadas: `fork(2)`, `wait(2)`,
4. Abra um terminal e execute o comando: `ps tree -c -p` Descreva o que está vendo.
5. Resolva http://wiki.inf.ufpr.br/maziero/doku.php?id=so:criacao_de_processos
6. Resolva <http://www.dei.isep.ipp.pt/~orlando/so2/processos.htm>
7. Familiarize-se com as chamdas: `execve(2)`, `clone(2)`,

Gerando uma árvore de processos

- Continuando o exercício da aula

Durante as aulas de 26 e 28 de março foram iniciadas as ideias de produzir um conjunto de processos para se gerar uma árvore.

Utilize a ferramenta `dot`, oriunda do pacote `graphviz` (possui pacote nas distribuições `debian` e `ubuntu`).

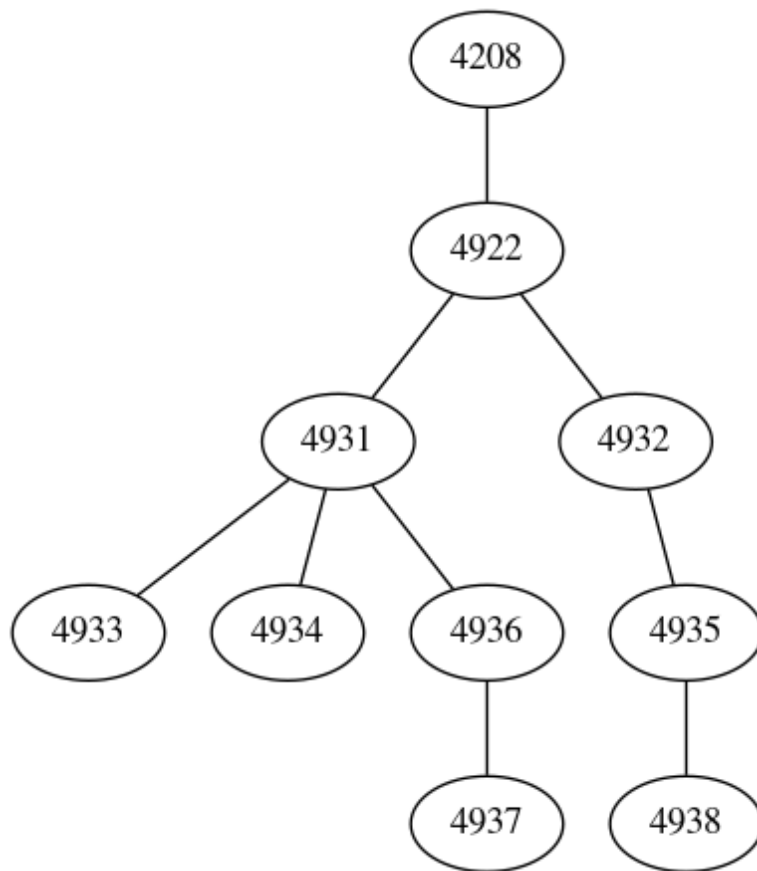
O comando `dot` gera um grafo (ou uma árvore) dada as ligações entre os nós. Por exemplo o arquivo abaixo (nomeado como `arvore-forks.dot`):

```
graph {  
4208 -- 4922;  
4931 -- 4933;  
4931 -- 4934;  
4922 -- 4931;  
4922 -- 4932;  
4931 -- 4936;  
4932 -- 4935;  
4935 -- 4938;  
4936 -- 4937;  
}
```

O arquivo acima pode ser renderizado utilizando o comando:

```
dot -t png arvore-forks.dot -o arvore-forks.png
```

Gerando uma imagem como abaixo:



Exemplo

Neste exemplo foi impresso o PID do PAI de um processo e o PID do processo, o dot cuidou para gerar as ligações.

Dado o código abaixo:

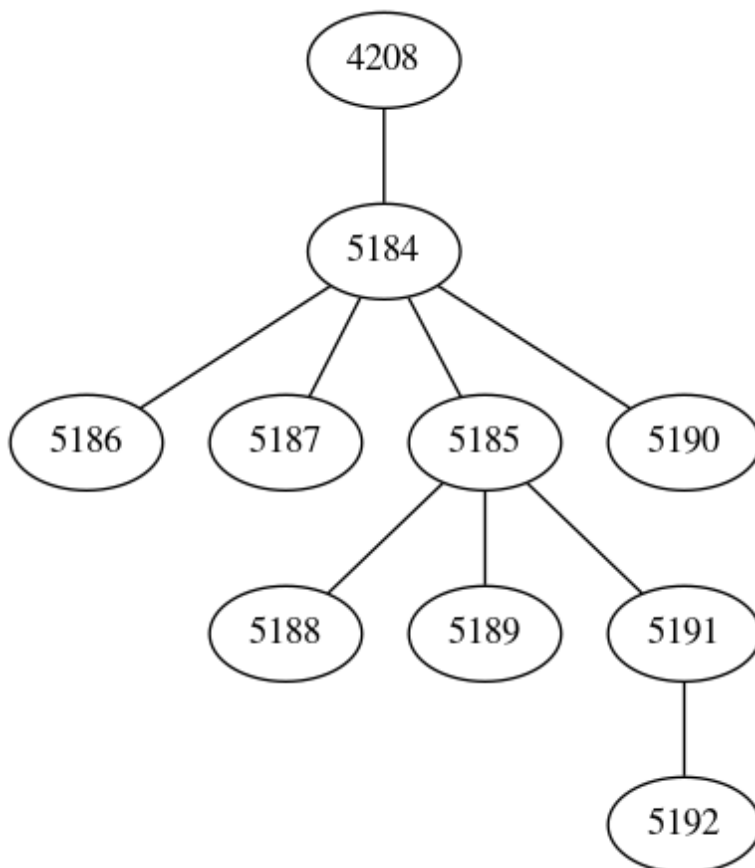
```
1  #include <stdio.h>
2  #include <unistd.h>
3  #include <sys/types.h>
4
5  int main(void)
6  {
7      if(fork(>0)
8      {
9          if(fork()==0)
10             if(fork()==0)
11                 fork();
12     }
13     else
14     {
15         if(fork())
```

```
16     if(fork())
17         if(fork())
18             ;
19         else
20             fork();
21     }
22     return 0;
23 }
```

1. Como é a árvore gerada no programa acima?
2. Modifique o programa acima para que ele imprima na tela o conteúdo de um arquivo para ser passado para o dot.



3. Faça um programa em c, que gere um conjunto de processos que tenha uma árvore semelhante como da ilustração abaixo:



Objetivo

Envie a solução do problema acima para o email *bruno.ribas EM unb.br* até o dia 7 de abril 23:59, com o assunto contendo a sua matrícula

1. Faça várias execuções do seu programa e verifique o comportamento do desenho da árvore, a ordem é sempre a mesma? Explique

Comunicando por FIFO - atividade de 4/4



1. Proponha um modelo de comunicação entre dois processos utilizando o tipo especial de arquivo FIFO, *mkfifo(3)* e sinais, *kill(2)*.
2. Faça um diagrama de como os processos devem se comunicar para enviar os sinais e as mensagens.

Envie a solução dos dois problema acima para o email *bruno.ribas EM unb.br* até o dia 7 de abril 23:59, com o assunto contendo a sua matrícula

Bruno Ribas - atividade para os dias 2 e 4 de abril