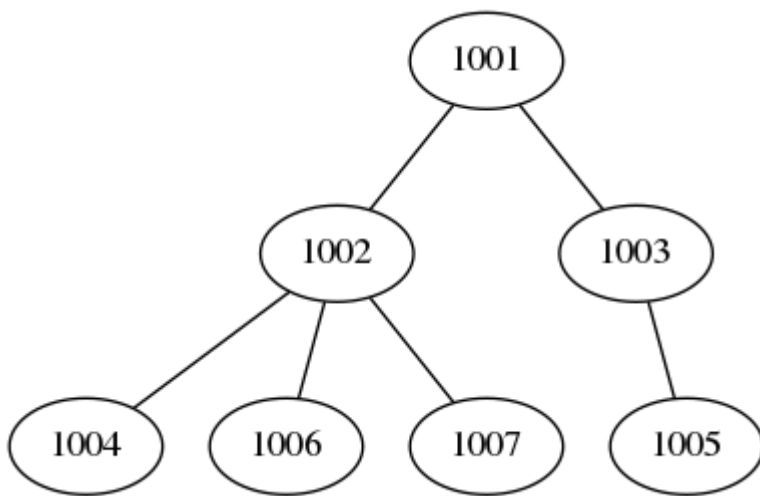


# Atividade 1 2019-2

Esta atividade faz parte da nota da disciplina e deve ser entregue até o dia 16 de setembro às 10h00 (não serão contabilizadas atividades entregues após este horário). Para entregar esta atividade envie um email para [ribas-fso-at1@naquadah.com.br](mailto:ribas-fso-at1@naquadah.com.br)

Esta atividade parte do pressuposto que o aluno já possui conhecimentos sobre `fork(2)`, `signal(2)` e `wait(2)`.

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



Além de montar a árvore citada acima, você deve resolver os problemas abaixo:

1. O Processo pai de todos, dado por 1001 no exemplo acima, precisa ler, da entrada padrão, uma String *S* de tamanho máximo de 100 caracteres.
  - Os processos filhos e netos de 1001 devem ser lançados antes ou depois de 1001 ler a string *S*?
2. O processo 1005 deve ser capaz de retornar o tamanho da string *S*, lida pelo processo 1001.
  - Como o processo 1001 faz para imprimir o tamanho da string *S*?
3. O processo 1006 deve alterar a string *S*, removendo todos os caracteres maiúsculos, e deve imprimir essa nova string a pedido do processo 1001
4. O processo 1004 deve remover os caracteres minúsculos de *S*
5. O processo 1007 deve remover as vogais de *S*

## Cronologia de eventos

---

O seu programa deverá seguir o seguinte roteiro de eventos:

- Ler a string *S* da entrada padrão
- Imprimir o tamanho da string *S*
- Imprimir a string *S* sem os caracteres maiúsculos
- Imprimir a string *S* sem as vogais
- Imprimir a string *S* sem os caracteres minúsculos

Perceba que a ordem de impressão é estrita, não podendo ser modificada.

O processo 1001 deve morrer apenas após a conclusão de todas as atividades. Os processos 1002 a 1007 podem morrer assim que cumprirem seu objetivo.

**ATENÇÃO** o processo pai (1001) não deve cumprir as tarefas que devem ser delegadas aos seus filhos e netos.

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