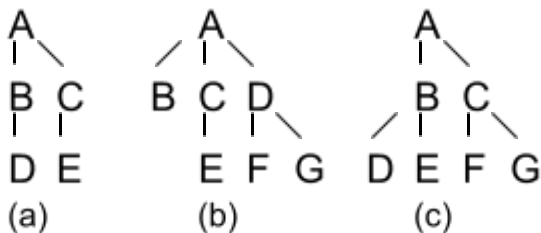




Universidade Federal de Santa Catarina
Centro Tecnológico de Joinville
Departamento de Engenharias de Mobilidade
EMB5632 - Sistemas Operacionais
Lista de exercícios 2 - Gerência de processos

1. Defina um sistema operacional.
2. Diferencie as quatro gerações da evolução dos computadores, caracterize cada uma delas.
3. O que caracteriza um deadlock?
4. Quais as funções básicas de um sistema operacional?
5. O que é uma chamada de sistema?
6. Escreva programas usando fork() que reproduzam as árvores abaixo:



O processo A deve executar printf("A") o processo B printf("B") e assim por diante.

7. Considere o código abaixo:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <unistd.h>
4 #include <sys/types.h>
5 #include <sys/wait.h>
6
7 int main(void)
8 {
9     pid_t f1, f2;
10
11     f1 = fork();
12     if (f1 != 0) {
13         f2 = fork();
14         if (f2 == 0) {
15             puts("P");
16             puts("F");
17         } else {
18             puts("T");
19             puts("S");
20         }
21     } else {
22         puts("C");
23         puts("H");
24     }
25     return 0;
26 }
```

- (a) Desenhe a árvore de Processos criada por este programa
- (b) Determine sete possíveis sequências que podem ser apresentadas pelo programa

8. Considere o código abaixo:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <unistd.h>
4 #include <sys/types.h>
5 #include <sys/wait.h>
6
7 #define N 3
8
9 int main(void)
10 {
11     pid_t f[N], w;
12     int status, i;
13
14     for (i = 0; i < N; i++) {
15         f[i] = fork();
16     }
17     sleep(10);
18     for (i = 0; i < N; i++) {
19         if (f[i] != 0)
20             w = waitpid(f[i], &status, 0);
21     }
22 }
23 return 0;
24 }
```

- (a) Quantos processos são criados pelo código acima, incluindo o pai de todos?
- (b) Descubra expressão para o número de processos criados em função de N

9. Considere o código abaixo:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <unistd.h>
4 #include <sys/types.h>
5 #include <sys/wait.h>
6
7 int main(void)
8 {
9     int n = 2;
10    fork();
11    if (fork() > 0)
12        n++;
13    n += 5;
14    if (fork() == 0)
15        n++;
16    printf("n = %d\n", n);
17    return 0;
18 }
```

- (a) Desenhe a árvore de processos criada por este programa
- (b) Mostre uma possível saída.