MC514—Sistemas Operacionais: Teoria e Prática

Lista de exercícios II

Não é necessário entregar esta lista!

1. Na *man page* sobre mutex locks existe o seguinte comentário:

   > **ASYNC-SIGNAL SAFETY**
   > The mutex functions are not async-signal safe. What this means is that they should not be called from a signal handler. In particular, calling `pthread_mutex_lock` or `pthread_mutex_unlock` from a signal handler may deadlock the calling thread.

   Você sabe explicar a razão deste aviso? Se estivéssemos trabalhando com locks recursivos, a thread poderia entrar em deadlock? Neste caso, seria seguro invocar `pthread_mutex_lock` dentro do tratador?

2. Um estudante de SO pensou em propor o conceito de "semáforo recursivo" semelhante ao mutex lock recursivo estudado. Neste caso, se a mesma thread invocasse `sem_wait` mais de uma vez, o semáforo seria decrementado apenas uma vez.

   ```
   sem_wait(semaforo_recursivo); /* O semáforo é decrementado apenas */
   sem_wait(semaforo_recursivo); /* uma vez para as duas chamadas    */
   ```

   Você acha que este comportamento teria alguma implementação interessante? Implementações que usam o conceito tradicional de semafóros poderiam utilizar este novo conceito sem problemas?

3. Escreva um trecho de código que utiliza a função `fork()` e gera a seguinte hierarquia de processos:

   ```
     A
    /|\
   B C D
     |
     E
   ```

4. Descreva vantagens e desvantagens de cada uma das três abordagens para um processo aguardar a morte de um processo filho: `wait` bloqueante, `waitpid` não bloqueante e tratamento do sinal `SIGCHLD`.

5. Suponha que um processo pai invoca a função `fork()` e imediatamente depois vai dormir com o comando `pause()`, que interrompe a execução de um processo até que este receba um sinal. Quando começa a executar, o filho envia um sinal SIGALRM para acordar o pai.

```
void trata_SIGALRM(int signum) {
  printf("Ai que sono! Queria dormir mais...\n");
}

int main() {

  if ((pid = fork()) != 0) {
    signal(SIGALRM, trata_SIGALRM); /* Instalação do tratador de sinal */
    pause();  /* Pai espera ser acordado pelo filho */
  }
  else
    kill (getppid(), SIGALRM); /* Filho envia sinal para acordar o pai */

  return 0;
}
```

   Descreva problemas que impediriam este código de funcionar corretamente. Você saberia escrever uma versão deste código que teria o comportamento desejado? (Consulte a função `sigsuspend()`.)

6. Suponha que um processo quer ser notificado quando um processo neto morre. Como você faria para implementar esta funcionalidade?