UFES - DEPARTAMENTO DE INFORMÁTICA

1ª. Prova de Sistemas Operacionais / Sistemas de Programação II

Período: 2012/1 – Data: 27/04/2012 - Profa. Roberta Lima Gomes

1) Considerando o diagrama de estados do UNIX, explique como ocorrem as seguintes transições (2,0)*:*
   a. New → Ready
   b. User Running → Kernel Running
   c. Ready → Stopped (Pronto Suspenso)
   d. Asleep (bloqueado) → Zombie

2) *Throughput* é o número de processos por hora que um sistema completa. *Turnaround time* é o tempo médio entre o momento que um processo é submetido e o momento em que ele termina. Comente como esses fatores são afetados se você adotar uma política de escalonamento do tipo *Shortest Job First*. (2,0)

3) Considere os dados da tabela: (2,0)

Processo	CPU Burst	Prioridade	Tempo de Chegada
P1	10	3	0
P2	5	1	4
P3	3	3	5
P4	1	4	6
P5	5	2	7

Desenhe 2 gráficos de Gantt: um para o algoritmo baseado em Prioridades (preemptivo) e outro para o algoritmo SRTF (*Shortest Remaining Time First*) e calcule o tempo médio de espera para os dois algoritmos.

4) Um grafo de precedência é um grafo dirigido, acíclico onde os nós representam atividades seqüenciais e onde um arco de um nó i para um nó j indica que a atividade i deve terminar antes que a atividade j possa começar. Desenhe o grafo de precedência do código abaixo em que cada nó representará uma instrução do programa (suponha que todas as chamadas execvp sejam bem sucedidas) : (2,5)

```
int f1, f2, f3; /* Identifica processos filho*/
int main(){
   printf("Alo do pai¥n");
   if ((f1 = fork())>0){
      execlp( "ps" , "ps" ,NULL);
      if (errno==0){
           printf("... vou  terminar¥n");
           exit(1);
      }
   }
   printf("Filho 1 criado¥n");
   if ((f2 = fork())==0){
      execlp( "ls" , "ls" ,NULL);
      if (errno==0){
           printf("... vou  terminar¥n");
           exit(1);
      }
  }
  printf("Filho 2 criado¥n");
  waitpid( f2,null,0);
  printf("Filho 2 morreu¥n");
  f3 = fork();
  execlp( "ps" , "ps" ,NULL);
  printf("Filho 3 criado¥n");
  waitpid( f3,null,0);
  printf("Filho 3 morreu¥n");
  exit();
}
```

5) No UNIX, como um processo PAI pode saber quando e como (por que) um processo FILHO terminou? Explique. (1,5)