**Trabalho 2 – Sistemas Operacionais**

Aluno: Otávio Malta Borges

Matrícula: 12011BSI291

**Ex4)**

Texto

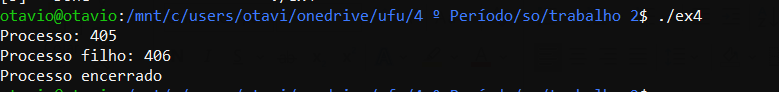
Descrição gerada automaticamente

Primeiramente é printado o PID do processo pai e posteriormente criado um fork(), o qual uma variável ‘pid’ receberá o retorno.

Caso o retorno seja -1, é printado uma mensagem de erro informando que não foi possível criar o processo filho.

Caso o pid for igual a 0 ( o processo filho foi criado), é printado o pid do processo o qual foi criado e sem seguida é finalizado com exit(0).

Caso o pid for maior que 0, significa que é o processo pai, portanto é chamado a função sleep(10) para manter o processo filho(o qual já se encerrou) no sistema por mais 10 segundos, sendo assim um Zumbi.

Saída: 

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Executando o comando os -l é possível ver os processos zumbis existentes (pid 395 e 398).

Terminando um processo com kill -9 (pid): Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

**EX5)**

É criado a estrutura do processo, com o nome, burst, turn around time, waiting time, status, chegada e um contador.

Texto

Descrição gerada automaticamente

É realizado então a abertura do arquivo ‘tarefas.txt’ e passado as informações para os processos. Além do nome e burst (os quais são pegos do arquivo), o status do processo é setado para 0 (pronto), setado a chegada (0, como pede o enunciado) e a variável ‘fim’ e ‘btAux’, as quais serão usadas no algoritmo Round Robin, são setados com o burst pego do arquivo para cada processo.

Texto

Descrição gerada automaticamente

É printado os processos e solicitado ao usuário a escolha entre os dois arquivos(FCFS e Round Robin).

Texto

Descrição gerada automaticamente

**FCFS:**

Um loop é criado, para acessar cada processo e então calculado o waiting time (tempo da ultima execução(i) – tempo de chegada do processo (p[k].chg)). O status é setado para 1 (finalizado) e o tempo da execução somado a variável ‘i’. O turn around time é calculado (waiting time + o tempo de execução – por não ser preemptivo).Os processos e seus estados são printados e a somados os TT e wt( para o cálculo da média). Após todos os processos e dados serem processados, é printado as médias.

Texto

Descrição gerada automaticamente

**Round Robin**

Primeiro é solicitado o quantum desejado. Logo é criado um loop enquanto houver algum processo pronto para ser executado.

O primeiro if analisa se é a primeira vez em que tal processo esta sendo executado, caso for, é setado o WT a partir da variável inicio, a qual se refere ao tempo inicio da execução do processo.

O próximo if verifica se o processo possui burst e se não ultrapassa o tempo do quantum. Caso isso ocorra, o tempo do fim do processo é somado com o burst (auxiliar, referente a quantidade atual do burst do processo, diferente do p.bt, o qual refere-se ao tempo inicial do burst do processo).

Caso o burst do processo for maior que a quantidade do quantum, o burst é decrementado e o tempo do processo somado ao quantum (para setar o tempo da duração do processo)

O próximo if verifica se o processo atual não possui mais burst e acabou de ser executado. A partir disso é setado o TT a partir da variável ‘fim’, correspondente ao tempo final do processo, o status é setado para 1 (finalizado) e a quantidade de processos decrementados.

**Texto

Descrição gerada automaticamente**

Por fim, após o loop, é printado os processos e somado as variáveis para o cálculo das médias.

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente