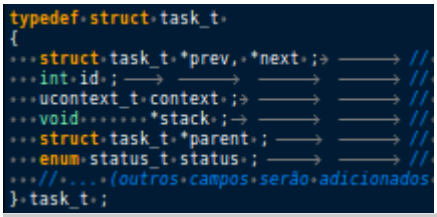

Carlos Maziero

Escalonamento por prioridades

Este projeto consiste em adicionar um escalonador baseado em prioridades com envelhecimento ao nosso sistema operacional.

As seguintes operações devem ser implementadas:

```
void task_setprio (task_t *task, int prio)
```

Ajusta a prioridade estática da tarefa `task` para o valor `prio` (que deve estar entre -20 e +20). Caso `task` seja nulo, ajusta a prioridade da tarefa atual.

```
int task_getprio (task_t *task)
```

Obtém o valor da prioridade estática da tarefa `task` (ou da tarefa corrente, se `task` for nulo).

Observações

- O escalonador deve usar prioridades no estilo UNIX (valores entre -20 e +20, com escala negativa).
- Para que o escalonador funcione corretamente, ele deve implementar um esquema de envelhecimento de tarefas (*task aging* com $\alpha = -1$). Caso contrário, sempre a mesma tarefa será escalonada para execução. O envelhecimento deve ser implementado **dentro do escalonador**.
- Ao ser criada, cada tarefa recebe a prioridade default (0).

Sua implementação deve funcionar com este código. A saída da execução deve ser **similar** a este exemplo (pequenos desvios são aceitáveis). Neste exemplo, observe que a tarefa `pang` executa com mais freqüência que a tarefa `peng`, e assim sucessivamente, o que indica claramente a influência das prioridades das tarefas no escalonamento.

Outras informações

- Duração estimada: 3 horas.
- Dependências:
  - Gestão de Tarefas
  - Dispatcher