

# Programação de Tarefas Periódicas

Dr. Osmar Marchi dos Santos

# Sinais (*Signals*)

- A criação de um Timer em POSIX permite gerar sinais em intervalos regulares de tempo
- É possível criar um software que captura sinais vindos do SO e que liberam a thread para execução
  - Assim, a thread fica em modo bloqueio (sem utilizar CPU) até que seja acordada pelo SO
- Para lidar com sinais é necessário utilizar a biblioteca `<signal.h>`
  - Sinais em POSIX para Tempo Real variam de SIGRTMIN (33) até SIGRTMAX (64) – 32 sinais por processo

# Sinais (*Signals*)

- Em especial, para capturar sinais utiliza-se:
- `sigset_t sinais; // Conjunto de sinais`
- `sigemptyset(sigset_t *S); // Limpa um conjunto de sinais`
- `sigaddset(sigset_t *S, int N); // Adiciona sinal N ao conjunto`

# Sinais (*Signals*)

- `int sigprocmask(int HOW, const sigset_t *S, sigset_t *OLDSET);`  
// Indica COMO (SIG\_BLOCK, SIG\_UNBLOCK) bloquear sinais
- `int sigwait(const sigset_t *S, int SIG);` // Espera a ocorrência de um sinal do conjunto de sinais. Ao capturar o sinal, SIG aponta para o sinal que foi capturado

# Sinais (*Signals*) – Exemplo

```
#include <signal.h>
#include <stdio.h>
#include <stdlib.h>

int main() {
    int sig;
    sigset_t s;
    sigemptyset(&s);
    sigaddset(&s, SIGINT);                // CONTROL-C
    sigprocmask(SIG_BLOCK, &s, NULL);
    for (;;) {
        sigwait (&s, &sig);
        printf("Recebeu sinal: %d\n", sig);
    }
    return 0;
}
```

// Buscar outros sinais POSIX e modificar o exemplo para capturar outros sinais

# Timers POSIX

- Possibilita gerar um sinal específico ao software no momento que um intervalo de tempo terminar (<time.h>):
- `timer_t timer; // Tipo timer`
- `timer_create(clockid_t C, struct sigevent *SE, timer_t *TID); // Cria um timer utilizando o tipo de relógio selecionado (CLOCK_REALTIME, CLOCK_MONOTONIC), repassa notificação para estrutura sigevent e cria timer`
- `timer_settime(timer_t timerid, int flags, const struct itimerspec *V, struct itimerspec *OV); // ID Timer e estrutura de tempo V indicando quando gerar o sinal (tempo chega em zero)`

# Periodicidade de Tarefas

- O código a disposição no Moodle apresenta o uso dos conceitos discutidos até o momento para implementar tarefas periódicas
- Instrumentar, compilar e executar o código para entender em detalhes o funcionamento do mecanismo utilizando sinais

# Exercício

- Criar um software com 10 tarefas periódicas (threads), com intervalos de 10 segundos para todas as threads. Cada thread executa por 1 segundo (usar *clocks* de CPU para threads) e depois espera pelo próximo período. Instrumentar o código para mostrar o comportamento das tarefas.