

Sistema de Monitorização e Controlo de uma Estufa

1 Introdução

Este trabalho tem como objectivo fundamental a familiarização por parte dos alunos com os aspectos de concorrência, sincronização e comunicação oferecidos por um sistema operativo (Unix/Linux). Os aspectos da interface aplicacional com o utilizador são relegados para segundo plano, sendo portanto simplificados.

A aplicação que se pretende concretizar – um sistema de monitorização e controlo – envolve os aspectos de comunicação entre o(s) utilizador(es) e o sistema de monitorização propriamente dito (constituído por diversos processos), bem como aspectos de sincronização entre os diversos componentes do sistema, e ainda o registo e consulta de um histórico sobre a evolução do sistema (interacção com o sistema de ficheiros).

2 Descrição geral do problema

A aplicação a programar (Sistema de Monitorização e Controlo de uma Estufa) é constituída por vários processos que comunicam entre si, e está dividida em três partes fundamentais: a interface com o utilizador (“**intuti**”), o sistema de monitorização e controlo propriamente dito (“**sismon**”), e o registo histórico (“**reghist**”) (ver Figura 1).

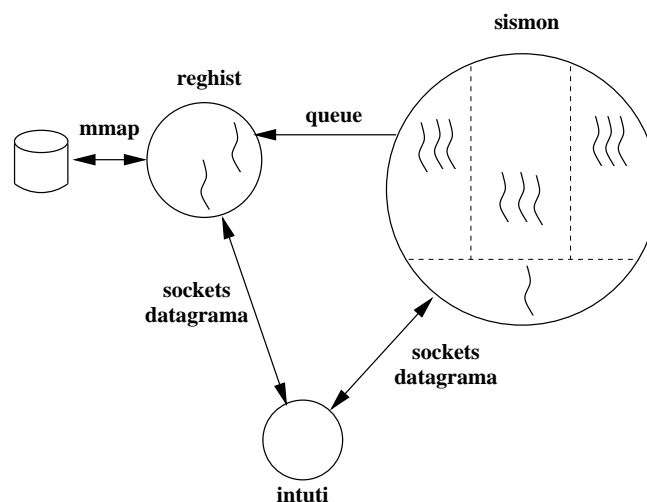


Figure 1: Arquitectura global do sistema de monitorização e controlo

A interface com o utilizador (“**intuti**”) permite efectuar comandos de consulta e configuração do sistema de monitorização, bem como consulta do registo histórico.

O sistema de monitorização e controlo (“**sismon**”) está organizado por sectores (NS sectores), existindo em cada sector 1 processo sensor (responsável pela leitura da temperatura e da humidade), 1 processo actuador (responsável pela abertura/fecho de janelas e sistema de rega) e 1 processo que representa o ambiente. Os processos sensores executam-se em ciclo, de forma periódica, recolhendo valores dos sensores (temperatura e humidade). Os valores recolhidos podem ser guardados de forma persistente, através do seu envio para o processo que faz o registo histórico, se esse envio estiver activo. O sistema de monitorização suporta ainda o diálogo com o(s) utilizador(es).

O registo histórico (“**reghist**”) recebe a informação proveniente do sistema de monitorização e guarda-a num ficheiro. Para além disso aceita comandos, da parte do utilizador, para consulta dos registos guardados.

Para que a aplicação global tenha a funcionalidade completa é necessário que os programas **sismon** e **reghist** estejam em execução. No entanto, deve ser possível suportar o funcionamento parcial de forma independente, ou seja, ter só a parte relativa ao **sismon** ou só a parte relativa ao **reghist**.

Para se poderem terminar de forma ordeira, e salvaguardando toda a informação relevante, tanto o **sismon** como o **reghist** devem aceitar uma ordem de terminação, que pode ser enviada a partir da aplicação **intuti**, e também através de um sinal **SIGTERM** enviado a partir da linha de comando (“shell”).

3 Interface com o utilizador – **intuti**

O programa responsável pela interface com o utilizador (**intuti**) suporta um conjunto de comandos para interagir quer com o sistema de monitorização propriamente dito (**sismon**), quer com o processo de registo histórico (**reghist**). São os seguintes, os comandos a disponibilizar:

Comandos Disponíveis		
comando	argumentos	descrição
cth	<i>s</i>	- consultar temperatura e humidade do sector <i>s</i> (0-todos)
cp	<i>s</i>	- consultar parâmetros (períodos, estado janelas/regagem) sector <i>s</i> (0-todos)
aj	<i>s</i>	- abrir janelas do sector <i>s</i> (0-todos)
fj	<i>s</i>	- fechar janelas do sector <i>s</i> (0-todos)
lr	<i>s</i>	- ligar rega do sector <i>s</i> (0-todos)
dr	<i>s</i>	- desligar rega do sector <i>s</i> (0-todos)
mps	<i>s p</i>	- modificar período proc.sensor do sector <i>s</i> (0-todos) (segundos)
mpa	<i>s p</i>	- modificar período proc.ambiente do sector <i>s</i> (0-todos) (segundos)
cala		- consultar limiares alarme temperatura/humidade (mínimos e máximos)
dala	<i>t T h H</i>	- definir limiares temperatura/humidade, min (<i>t, h</i>) e max (<i>T, H</i>)
cer		- consultar estado envio registos para histórico
aer		- activar envio de registos para histórico
der		- desactivar envio de registos para histórico
tsm		- terminar processo sistema de monitorização (sismon)
lreg	<i>s [t1 [t2]]</i>	- listar registos sector <i>s</i> (0-todos) entre <i>t1</i> (dd/mm/aaaa hh:mm:ss) e <i>t2</i>
trh		- terminar processo de registo histórico (reghist)
sos		- mostrar comandos disponíveis
sair		- sair da interface utilizador (intuti)

No primeiro grupo de comandos a interacção é feita com o sistema de monitorização (**sismon**), e no segundo grupo de comandos a interacção é feita com o processo de registo histórico (**reghist**). Em qualquer dos casos existe uma mensagem de resposta, ficando o processo bloqueado à sua espera (interface síncrona). Deve, no entanto, ser prevista a hipótese de impossibilidade de comunicação, com o correspondente desbloqueio ("timeout"). No caso em que o comando especificado não seja executado com sucesso remotamente, a mensagem de resposta terá um código de erro.

Na especificação do intervalo temporal ($[t1\ t2]$) cada instante é dado no formato "**dd/mm/aaaa hh:mm:ss**", sendo no entanto facultativos: a ausência de $t1$ e $t2$ significa todos os registos; a ausência de $t2$ significa de $t1$ até ao fim.

4 O sistema de monitorização – **sismon**

Como se disse atrás, o sistema de monitorização está organizado em sectores (**NS** sectores). Em cada sector existe um processo sensor, um processo actuador e um processo ambiente. Cada processo corresponde a uma "thread" (pthread). As "threads" de uma mesma classe são idênticas entre si (mesmo código), distinguindo-se apenas pelos parâmetros que lhe estão associados. Os processos de cada sector são conhecidos por um identificador (correspondente ao respectivo sector – valores 1..NS) que lhes é atribuído no momento da criação.

Os valores da temperatura e humidade em cada sector são representados por duas variáveis que serão acedidas pelas várias entidades que as manipulam: processo sensor para ler; processo ambiente para alterar (valores iniciais da temperatura e humidade: **TINI**, **HINI**). Existem também por cada sector duas variáveis que indicam o estado (**ABERTO/FECHADO**) do sistema de rega e das janelas (inicialmente fechados). O estado destas variáveis será modificado pelo processo actuador, quando tal for solicitado. E serão também acedidas pelo processo ambiente.

Um processo ambiente simula as alterações da temperatura e humidade provocadas pelo ambiente no respectivo sector. Altera periodicamente (período inicial igual a **PAMB**) as variáveis correspondentes a essas grandezas de acordo com os seguintes algoritmos.

Temperatura:

- determina de forma aleatória (função **random**) uma componente variável (valores -1, 0, +1);
- utiliza uma componente fixa (+1 ou -1, com alternância a cada **NCICL** ciclos);
- utiliza uma componente fixa (+1 ou -1, consoante a janela esteja fechada ou aberta);
- adiciona o valor resultante à variável temperatura.

Humidade:

- determina de forma aleatória (função **random**) uma componente variável (valores -1, 0, +1);
- utiliza uma componente fixa (+1 ou -1, com alternância a cada **NCICL** ciclos);
- utiliza uma componente fixa (+1 ou -1, consoante o sistema de rega esteja ligado ou desligado);
- adiciona o valor resultante à variável humidade.

Um processo sensor recolhe periodicamente (período inicial igual a **PSEN**) os valores da temperatura e da humidade. A última ocorrência desses valores, juntamente com a data/hora de ocorrência, é mantida em memória local para poder ser consultada com o comando **cth**. Para além disso, no caso de algum dos valores recolhido ser **diferente** do valor anterior, e a opção de envio de registos estar activa, será criado um registo (ver registo **reg-t**) e enviado para o processo de registo histórico (**reghist**).

Aquando da leitura da temperatura e da humidade, o processo sensor deve também verificar se está perante uma situação de alarme: temperatura ou humidade menor que o valor mínimo, ou temperatura ou humidade maior que o valor máximo (valores iniciais **TMIN**, **HMIN**, **TMAX** e **HMAX**). Em caso afirmativo afixará localmente uma mensagem de alerta.

Para além dos processos referidos acima, o sistema de monitorização tem ainda a capacidade de dialogar com a interface utilizador para execução dos comandos referidos anteriormente.

O programa **sismon** é activado de forma independente, sendo responsável pelas inicializações necessárias ao correcto funcionamento da aplicação (criação dos vários objectos de comunicação e sincronização). Caso o processo de registo histórico não esteja ainda disponível, e seja activado o envio de registos, também deverá ser criado a partir deste programa.

5 O registo histórico – **reghist**

O programa **reghist** pode ser activado de forma independente, ou a partir do programa **sismon**. É responsável por guardar em memória persistente (ficheiro **DADOS**) os registos que lhe são enviados pelo sistema de monitorização. Para além disso, deve ainda permitir a consulta de registos feita a partir da interface utilizador. O ficheiro **DADOS** deve ser mantido entre as várias execuções de **reghist**.

Este programa possui 2 “fios-de-execução” (“threads”): um dedicado à comunicação com a interface utilizador (por exemplo o programa principal); outro para receber a informação vinda do sistema de monitorização **sismon**.

Para evitar um crescimento desmesurado do ficheiro, existirão, num dado instante, até um máximo de **NREG** registos. A partir deste valor, os novos registos sobrepõem-se aos anteriores, funcionando o ficheiro como um “buffer-circular”.

A gestão do ficheiro pode ser efectuada quer utilizando as primitivas mais tradicionais de acesso a ficheiros (**open**, **read**, **write**, ...), como mapeando o ficheiro em memória (**mmap**) (aconselhado).

6 Objectos de comunicação e sincronização

A sincronização entre as várias “threads” existentes num processo, como por exemplo o controlo do acesso destas a eventuais estruturas de dados partilhadas, deve ser feito utilizando semáforos POSIX (**sem_init**, **sem_post**, **sem_wait**, ...), ou as próprias primitivas associadas às “pthreads” (**pthread_mutex_lock**, **pthread_mutex_unlock**, ...).

No caso de haver necessidade de sincronizar processos com espaços de endereçamento distintos, podem ser utilizados os mecanismos do IPC Sistema V.

Na comunicação entre a interface utilizador, quer com o sistema de monitorização quer com o

registo histórico, devem ser usados “sockets” (sockets unix datagrama).

A comunicação entre os vários processos (“threads”) do sistema de monitorização (**sismon**) e o processo que faz o registo histórico deve ser efectuada por mensagens (*message queues*), preferencialmente utilizando a interface POSIX (`mq_open`, `mq_send`, `mq_receive`, ...), ou, em alternativa, utilizando as primitivas IPC do Sistema V (`msgget`, `msgctl`, `msgsnd`, `msgrcv`, ...). Esta comunicação, do sistema de monitorização com o processo de registo histórico, tem uma interface assíncrona, não havendo qualquer mensagem de resposta.

O programa **sismon**, do ponto de vista da comunicação com a interface utilizador (socket unix datagrama) é conhecido pelo nome global **SISMON**.

O programa **reghist**, do ponto de vista das comunicações, é conhecido pelos nomes globais:

REGS - para a interface do utilizador (nome socket unix datagrama);

REGQ - para o sistema de monitorização (nome queue POSIX ou chave caixa correio).

7 Desenvolvimento do projecto

No desenvolvimento do projecto, aconselha-se a utilização de uma estrutura modular, com testes faseados.

Na interface com o utilizador, **não** se pretende nada de muito complexo (não é esse o objectivo fundamental). Para simplificar a sua concretização, os alunos **devem** utilizar o interpretador de comandos rudimentar (`cmd`), que é fornecido (ver página da disciplina).

8 Estruturas de dados e constantes

Na concretização do trabalho considere os seguintes valores para as constantes referidas anteriormente:

```
#define NS      3                /* numero de sectores a controlar */
#define PSEN    5  /* período inicial dos processos sensores (em segundos) */
#define PAMB   10  /* período inicial dos processos ambiente (em segundos) */
#define TINI   25                /* temperatura inicial dos sectores (em graus) */
#define TMIN  -100  /* temperatura minima inicial para alarme (em graus) */
#define TMAX  +100  /* temperatura maxima inicial para alarme (em graus) */
#define HINI   50                /* humidade inicial dos sectores (em %) */
#define HMIN    0                /* humidade minima inicial para alarme (em %) */
#define HMAX  100                /* humidade maxima inicial para alarme (em %) */
#define NCICL  12                /* numero de ciclos para alternancia ambiente */

#define NREG    300              /* numero maximo de registos no ficheiro */
#define DADOS   "HISTORICO.LOG" /* ficheiro com registo historico */

#define SISMON  "/tmp/SISMON"    /* nome do sistema de monitorização (socket) */
#define REGS    "/tmp/REG"       /* nome do registo historico (socket) */
#define REGQ    "/REGQ"          /* nome do registo historico (queue) */
// ou #define REGQ  0x52          /* ('R') nome (chave) do registo historico (msg) */
```

```
typedef struct reg_s {
    struct timespec t;
    int s;
    int t;
    int h;
} reg_t;

/* estrutura de um registo */
/* estampilha temporal */
/* identificador do sector */
/* valor da temperatura */
/* valor da humidade */
```

9 Entrega do Trabalho

Este trabalho tem uma **meta intercalar na semana de 4 de Abril** (no horário de laboratório (5f-7/04, 6f-8/04)). Deverão mostrar a funcionar, **pelo menos**, uma versão rudimentar do sistema que inclua:

- comunicação entre a interface do utilizador e o sistema de monitorização (comandos para consultar temperatura/humidade e parâmetros, modificar períodos e estado do sistema, definir alarmes, terminar processo).

O **trabalho final** deve ser entregue até ao dia 22 de Abril de 2022. O material a entregar consiste numa cópia em formato digital (ficheiro ZIP) de todos os programas desenvolvidos e respectivo “makefile”, e um pequeno relatório (3 ou 4 páginas) descrevendo as principais estruturas de dados usadas, bem como os aspectos relativos à comunicação e sincronização entre as várias entidades concorrentes existentes. Todos estes elementos devem ser identificados com o número do grupo de laboratório e alunos que o compõem.

As **visualizações e discussões dos trabalhos** serão efectuadas na **semana de 25 de Abril** (em horários a definir) com base no trabalho entregue.