



Universidade da Beira Interior
Sistemas Operativos 2018
Trabalho de Grupo

Titulo/ Title: Logging Server
Duração/Duration 4 Semanas/4 Weeks

V1: 6/Maio 2018

Objetivos

Os objetivos deste trabalho são a aprendizagem de programação entre processos / threads usando pipes e filas de mensagens e possivelmente a multiprogramação usando threads e/ou processos. Este projeto será desenvolvido por equipas de 2 ou até 3 alunos de preferência do mesmo turno prático.

Requisitos

Desenvolvimento dum servidor de “logging” e aplicações associadas. Um servidor de logging recebe dados enviados pelas aplicações clientes para processamento, armazenamento e divulgação. Os requisitos funcionais são especificadas numa maneira geral em baixo. Não será dada uma especificação mais detalhada dos requisitos dos projetos para que haja liberdade de cada grupo criar algo diferente. Podem usar o fórum do Moodle para discutir outras funcionalidades.

Develop a logging server and associated applications. A logging server receives data send by client applications for processing, storage and dissemination.

Avaliação

- Entrega do Projeto Até Dia 3 de Junho. Deadline June 3
- Apresentação do Projeto – Na ultima semana (ou seguinte) das aulas (XXXX Junho)
- Grupos 2/3 Pessoas. Do mesmo TURMO das aulas praticas.

Entrega do Trabalho

- O trabalho consistirá no conjunto de diretórios e ficheiros de código (.c, .h etc), Makefiles (obrigatório!) que descreve o processo de compilação) e um ficheiro chamado “ProjetoSO18.xxx (.pdf, .doc ou .tex)” que contém os nomes/números dos autores e uma breve descrição do projecto realizado (Max.5 Páginas A4). Não é necessário um relatório detalhado, apenas os objetivos alcançados, os objetivos atingidos, pontos interessantes.
- Deverá também imprimir o relatório e entregar em papel.
- O trabalho deverá ser arquivado num ficheiro (tar ou winzip) e entregue eletronicamente via um upload link no Moodle.
Para quem utilize “git” podem simplesmente arquivar a pasta do projeto git e entregar este !
- Deverá imprimir o ficheiro “ProjectoSO18.xxx” e entrega-lo durante a apresentação e defesa do seu projeto.

Referências BIBLIOGRÁFICAS

- Fichas das Aulas Praticas - IPC - Comunicação entre processos 1ª parte
- Advanced Programming in the UNIX Environment (Livro Disponível no Moodle – podem fazer download do ficheiro ou aqui <https://tinyurl.com/y99hyodq>) Book Site . <http://www.apuebook.com/>

Logging Server: Description and Functional Requirements

There are three types of program.

- A logging server, client applications that send data to this server .
- A logging console application to interact with the server.
- An administration console, or console application .

Logging Server and Client Applications

- The “Logging Server Program” receives log data from client applications.
- The server should run as a background process . Only one instance of the server should be permitted to run, however the server can use multiprocessing (fork) or threading (thread) to create processes or threads that deal with the separate client sessions.
- The server should be run as a background/daemon/service i.e the user who starts the server should not need to be logged in to the computer.
- The group must provide some simple client applications to send data to the server
- Client applications send logging data to this server application via messages queues.
- Client applications will send data with one of two types of formats, logging data and requests for control or information purposes

Format 1 : Logging Data: The should contain information such as: PID, Program Name, Date/Time, Log Level, Session Number, Session Sequence Number, Message(Logging data)

Format 2 : Requests: Client applications may send and receive information for certain requests.

1-Request to start a new session and return of session ID or refusal.

2-Message to Terminate current session.

Optional/Additional Possibilities.

- Does the sever accept logging data in binary or only text format ?
- What is the frequency rate (data per second) that the server can process ?
- Specify which users may access the logging data etc. Output Formats etc

Notes. It is also possible to define a simple API inside a static/dynamically compiled library that client programs could use !

The Console Application to communicate with the server.

- Communication between the server and the console application will be via names pipes. This will permit only one client application to access the sever.
- The console application has two types of functionality (1) controlling the server and (2) accessing the logged data
- The console application should be able to start/stop/restart/pause the server. It should also be able to query the server about number of clients with open sessions and other useful information. It could also terminate sessions etc.
- The console application should also be able to request reports and access the logged data. This type of access could in fact be completely separate from the control application and use sockets for inter process communication – but this is more complex!)
- The console application should be password protected (either pass the password via the command line or inside the application) that is valid for a given time period (five minutes)