



DEPARTMENT OF AUTOMATION

PROF FRANCESCO QUAGLIA - SISTEMI OPERATIVI AVANZATI

TAG-based data exchange

Author:
Emanuele Alfano

Abstract

Lo scopo del sistema è di mettere in comunicazione un numero imprecisato di processi all'interno dello stesso kernel, e permettere lo scambio istantaneo di pacchetti tra 1 writer e i reader in attesa su di un canale.

La logica del sistema segue lo schema publisher-subscriber, dopo un subscriber, una volta ottenuto ciò che desidera, deve manualmente accodarsi nuovamente al publisher.

Le richieste, in dettaglio, del progetto sono reperibili al link: [SOA-prg](#)

Date
27/04/2021

Contents

List of Figures	i
List of Tables	i
1 Introduction	1

List of Figures

List of Tables

1 Introduction

Lo scopo del sistema è di mettere in comunicazione un numero imprecisato di processi all'interno dello stesso kernel, e permettere lo scambio istantaneo di pacchetti tra 1 writer e i reader in attesa su di un canale.

All'arrivo di un Writer, tutti i reader in attesa su questo canale vengono svegliati e ricevono il dato trasmesso dal Writer.

Il sistema è organizzato in 2 livelli di ricerca:

1. Tag-level search
Nel codice ci si riferisce a lui col termine "stanza", ne possono esistere fino a 256 di default, ma è possibile variare a run-time questo limite facendolo crescere a piacere, o decrescere fino al numero di stanze attualmente aperto, e comunque non meno di 256.
2. Level-level
Ogni stanza, possiede a sua volta 32 sotto livelli dove effettivamente i reader e writer parlano

Segue che nella stessa stanza(tag) è possibile svolgere in parallelo più conversazioni, e far accodare diversi thread su diversi topic, fino al limite di 32 topic per stanza.

In aggiunta a queste 2, le stanze possono possedere un "nome" pubblico, detto key, che permette di indicizzarle e farle trovare anche ad altri processi semplicemente cercando la key corrispondente.

Per interagire con il sistema sono state implementate 4 system-call "rubare" alla sys-call table dalle funzioni che puntavano alla `sys_ni_syscall`, ovvero quelle ancora implementate nel kernel.

1. `int tag_get(int key, int command, int permission)`
Per mezzo di questa sys-call è possibile creare una stanza pubblica o privata (permission) rispetto ai thread di altri processi, e scegliere se deve essere indicizzata globalmente o meno (key)
2. `int tag_send(int tag, int level, char* buffer, size_t size)`
Questa sys-call permette a un writer di pubblicare nella stanza desiderata al livello voluto
3. `int tag_receive(int tag, int level, char* buffer, size_t size)`
Questa sys-call mette un reader in attesa nella stanza al livello voluto finché un writer non la risveglia
4. `int tag_ctl(int tag, int command)`
Con quest'ultima syscall è possibile inviare dei comandi all'intera stanza, nella fattispecie la chiusura e la wake-up-all

Per avere più dettagli sulle richieste delle interfacce si faccia riferimento alla pagina ufficiale del progetto: [SOA-prg](#)
