

Sistemas Operativos II

Época Normal

1^a Chamada¹

15 de Julho de 2005

Duração: 2h30m

I

1. Explique as vantagens e desvantagens da utilização de um Log-structured File System (também conhecidos por Journaling FS ou sistema de ficheiros “baseado em diário”) face a um sistema de ficheiros tradicional. Apresente um caso prático em que recomendaria vivamente a sua utilização. Justifique.

2. Estude a função `int vota(int v)` do grupo seguinte. Se, em vez de threads e variáveis de condição, lhe fosse pedida uma solução baseada em processos e semáforos, essa solução seria mais fácil ou difícil? Justifique.

II

Construa uma função `int vota(int v)` que permite votar numa de entre 10 opções e apenas devolve um resultado final quando houver uma opção vencedora. Esta situação verifica-se logo que haja pelo menos 4 votos e uma das opções registre mais de 1/3 dos votos expressos. Esta função funciona num ambiente multi-threaded, e deve ser implementada com recurso a primitivas threads Posix.

III

Pretende-se desenvolver um programa que reencaminhe informação recebida num porto local para um porto de uma máquina remota. O programa deverá aceitar múltiplas conexões, reencaminhando a informação recebida em cada uma delas após T segundos. Para este efeito, será estabelecida uma conexão com o porto remoto apenas no momento do reenvio. Apresente o código fonte dum programa que cumpra estes objectivos recorrendo às primitivas de sockets e threads Posix (assuma que todos os parâmetros necessários serão especificados como argumentos da linha de comando).

¹Cotação — 7+6+7

Protótipos das chamadas ao sistema relevantes

Sistema de Ficheiros

- `int open(const char *pathname, int flags, mode_t mode);`
- `int creat(const char *pathname, mode_t mode);`
- `int close(int fd);`
- `int read(int fd, void *buf, size_t count);`
- `int write(int fd, const void *buf, size_t count);`
- `int pipe(int filedes[2]);`
- `int dup(int oldfd);`
- `int dup2(int oldfd, int newfd);`

Threads POSIX

- `int pthread_create(pthread_t *threadid, const pthread_attr_t *attr, void *(*start_func)(void *), void *arg);`
- `void pthread_exit(void *status);`
- `int pthread_join(pthread_t threadid, void **status);`
- `int pthread_detach(pthread_t threadid);`
- `int pthread_mutex_init(pthread_mutex_t *mp, const pthread_mutexattr_t *attr);`
- `int pthread_mutex_lock(pthread_mutex_t *mp);`
- `int pthread_mutex_unlock(pthread_mutex_t *mp);`
- `int pthread_mutex_destroy(pthread_mutex_t *mp);`
- `int pthread_cond_init(pthread_cond_t *cond, const pthread_condattr_t *attr);`

- `int pthread_cond_wait(pthread_cond_t *cond, pthread_mutex_t *mutex);`
- `int pthread_cond_signal(pthread_cond_t *cond);`
- `int pthread_cond_broadcast(pthread_cond_t *cond);`

Sockets BSD

- `int socket(int domain, int type, int protocol);`
- `int bind(int s, const struct sockaddr *name, int namelen);`
- `int listen(int s, int backlog);`
- `int accept(int s, struct sockaddr *addr, int *addrlen);`
- `int connect(int s, struct sockaddr *name, int namelen);`
- `int close(int s);`
- `int read(int fd, char *buffer, size_t len);`
- `int write(int fd, const char *buffer, size_t leng);`
- `u_long htonl(u_long hostlong);`
- `u_short htons(u_short hostshort);`
- `u_long ntohl(u_long netlong);`
- `u_short ntohs(u_short netshort);`
- `unsigned long inet_addr(const char *cp);`
- `struct sockaddr_in { short sin_family; u_short sin_port; struct in_addr sin_addr; char sin_zero[8]; };`