

UXP Zadanie 1

Początek realizacji: 09.04.2015 Projekt wstępny: 23.04.2015; Koniec realizacji: 11.06.2015
punktacja: 0 - 50 p.

Treść: Napisać wieloprocesowy system realizujący prosty system plików (w przestrzeni użytkownika, nie jądra). System powinien udostępniać następujące funkcje (zachowanie analogiczne do systemowych odpowiedników)

```
int simplefs_open(char *name, int mode);
int simplefs_unlink(char *name);
int simplefs_mkdir(char *name);
int simplefs_creat(char *name, int mode);
int simplefs_read(int fd, char *buf, int len);
int simplefs_write(int fd, char *buf, int len);
int simplefs_lseek(int fd, int whence, int offset);
```

System powinien być zrealizowany w jednym pliku | obszarze pamięci współdzielonej zachowującym się jak uproszczony dysk logiczny, tj. powinien on zawierać: tablicę i-node-ów, katalog główny, listę wolnych bloków, obszar na pliki. Należy zaimplementować tylko pliki zwykłe i katalogi.

Implementacja powinna pozwalać na jednoczesny dostęp wielu procesów, tak więc należy zrealizować wykluczanie i wielodostęp, nie na poziomie całego mini-systemu plików, ale możliwie „drobno-ziarniście”, tj. na poziomie poszczególnych obiektów (katalogów, tablicy i-node-ów). Wykluczanie zrealizować przy pomocy: funkcji fcntl() (zajmowanie rekordów), semaforów, kolejek komunikatów, potoków FIFO (z centralnym procesem serwerem).

Dodatkowe założenia i wymagania :

- Kontrolę dostępu na poziomie U/G/O należy pominąć, ale zróżnicować prawa R/W, można pominąć prawo X dla katalogów
- można pominąć flagi: APPEND, TRUNC, NODELAY, itp.
- można przyjąć statyczny rozmiar: katalogu głównego, pod katalogów i tablicy i-node-ów (bez relokacji)
- metryczka pliku może być maksymalnie uproszczona (nie musi odzwierciedlać struktury stat)
- można założyć ciągłą alokację z opcjonalnym mechanizmem defragmentacji

Miejsce realizacji: sprzęt własny, lab. 9 - Linux, lub inne dysponujące odpowiednią liczbą komputerów z systemem Unix/Linux

Warunki zaliczenia: Program *musi* być napisany w języku C/C++ wykonującym się w środowisku Unix (Java, Windows i inne rozwiązania nie są dozwolone). Zaliczenie zadania odbywa się na podstawie projektu wstępnego (10p), sprawozdania (20p) oraz pokazu działającego programu (20p – 10p. ogólna jakość produktu, 10p. jakość kodu) - w godz. konsultacji lub innym uzgodnionym terminie. Opóźnienie powoduje obniżenie punktacji. Niedostarczenie projektu wstępnego i/lub sprawozdania (opóźnienie 2 tyg. lub nie dostarczenie przed końcem semestru) powoduje niezaliczenie zadania i przedmiotu.

Instrukcje dot. realizacji projektu:

Co powinien zawierać projekt wstępny (3-4 strony)

Sprawozdanie musi być dostarczone osobiście - "na papierze", postać elektroniczna nie jest akceptowana i nie będzie brana pod uwagę. Dostarczone sprawozdanie może, ale nie musi być od razu omówione, ocena ze sprawozdania wstępnego zostanie wystawiona w ciągu tygodnia od dostarczenia, w przypadku oceny ≤ 8 p. konieczne jest omówienie i poprawienie sprawozdania.

Sprawozdanie wstępne powinno zawierać:

1. treść zadania
2. interpretację treści zadania
3. krótki opis funkcjonalny – "black-box" - proponowane API
4. opis i analizę poprawności zaproponowanych **protokołów komunikacyjnych** (wskazane - z rysunkiem)
5. planowany podział na moduły/procesy i strukturę komunikacji między nimi (być może z rysunkiem), w tym koncepcję realizacji **współbieżności**
6. zarys koncepcji implementacji (najważniejsze funkcje, algorytmy, obiekty komunikacyjne)

Co powinien zawierać projekt ostateczny (6-10 stron)

1. treść zadania
2. interpretację treści zadania
3. pełen opis funkcjonalny – "black-box" - proponowane API
4. podział na moduły i strukturę komunikacji między nimi (silnie wskazany rysunek)
5. opis najważniejszych rozwiązań funkcjonalnych wraz z uzasadnieniem (opis protokołów, kluczowych funkcji, itp.)
6. szczegółowy opis interfejsu użytkownika
7. postać wszystkich plików konfiguracyjnych, logów, itp.
8. Opis wykorzystanych narzędzi, itp.
9. Opis metodyki testów i wyników testowania

Uwaga:

- Sprawozdanie wstępne przekazane zostaje w formie pisemnej, nie jest wymagana obecność całego zespołu przy jego przekazaniu
- Konsultacje w trakcie realizacji projektu nie wymagają obecności całego zespołu.
- Sprawozdanie końcowe i pokaz funkcjonowania musi odbywać się w obecności całego zespołu, pozostałe warunki formalne – patrz spr. wstępne.
- opisywanie ogólnie znanych rozwiązań (np. API SYSV IPC) jest niepotrzebne i niewskazane
- b. ważne jest precyzyjne opisanie obsługi **sytuacji wyjątkowych i reakcji na błędy**
- b. ważne jest szczegółowe opisanie przeprowadzonych testów – UWAGA: testy nie mają na celu wykazania, że program **działa** poprawnie. Test ma na celu wykazanie, że program **nie działa** poprawnie!
- kod źródłowy proszę przysyłać wyłącznie w formacie tar.gz na adres: GBlinowski@ii.pw.edu.pl dopiero po zaaprobowaniu programu i sprawozdania