Programowanie systemowe I

współbieżne

Sprawozadnie 4

Wprowadzenie:

Celem zajęć było zapoznanie się z kolejkami komunikatów, pamięcią współdzieloną oraz z semaforami. Polecenia przydatne na tych zajęciach to: msgget(), msgctl(), sgsnd(), msgrcv(), shmget(), shmat(), shmdt(), hmctl(), semget(), semop(), semctl(), raise().

Poniżej znajdują się rozwiązania zadań 4 oraz 6, które polegały na napisaniu programów zgodnych z podaną treścią, lub odpowiedzi na zadane pytanie. Programy były napisane w edytorze tekstu vim, a skompilowane przy użyciu kompilatora gcc.

Zadanie 4

Zadanie polegało na utworzeniu bloku pamięci współdzielonej wykorzystując jako klucz swój numer indeksu, a następnie napisanie 2 programów. Pierwszy zapisuje tekst do pamięci współdzielonej a drugi go odczytuje.

Program 1 tworzy pamięć współdzieloną poleceniem shmget, następnie podłącza segment pamięci do przestrzeni adresowej procesu poleceniem shmat i zapisuje do pamięci tekst za pomocą strcpy. Na koniec pamięć zostaje odłączona poleceniem shmdt.

Program 2 otwiera istniejący segment pamięci poleceniem shmget, a następnie podłącza segment pamięci współdzielonej do przestrzeni adresowej procesu poleceniem shmat. Po tym zawartość danych z pamięci współdzielonej jest wypisywana polecniem printf. Na koniec pamięć współdzielona zostaje odłączona od przestrzeni adresowej procesu poleceniem shmdt oraz segment pamięci współdzielonej zostaje usunięta poleceniem shmctl.

Zadanie 6

1. Semafor jest uważany za otwarty w momencie kiedy jego wartość jest większa bądź równa 0.
2. W momencie gdy zmienna x została zmieniona w procesie macierzystym już po wywołaniu funkcji fork to jej zmiana nie będzie uwzględniona w procesie potomnym. Wynika to z tego, że nowo utworzony proces potomny ma własną przestrzeń adresową, w której x jest zmienną lokalną.
3. Skutkiem wywołania wait(NULL) przez proces macierzysty będzie zablokowanie procesu macierzystego, aż proces potomny nie skończy swojego działania. W tym programie przez wywołanie wait(NULL) zatrzyma polecenie wypisujące x2, aż do momentu zakończenia procesu potomnego przez co x2 wypisze się ostatnie.
4. Na początku proces potomny jest zablokowany. Proces macierzysty wypisuje 4 oraz podnosi semafor o 5, czym aktywuje proces potomny, a następnie oczekuje na zakończenie procesu potomnego. Po tym proces potomny zmniejsza semafor o 2 i wypisuje 6. Następnie semafor znowu zostaje zmniejszony o 2 i wynosi 1. Proces potomny wypisuje 10 i zakańcza się, czym aktywuje proces macierzysty. Po tym proces macierzysty wypisze 4.

Kolejność wypisania wynosi:  
x1=4

X3=6

X4=10

X2=4

Dawid Gorszka, nr indeksu: 162375