SYSTEMY WEJŚCIA-WYJŚCIA

Sprzęt wejścia-wyjścia

Port (port) - punkt łączący urządzenie z komputerem.
Szyna, magistrala (bus) - wspólna wiązka przewodów, z których korzystają urządzenia.
Łańcuch (daisy chain) - układ szeregowy urządzeń - wtedy dzielony dostęp bezpośredni.
Sterownik (controller) - zespół układów elektronicznych, służący do kierowania pracą portu, szyny bądź urządzenia. Komunikacja między sterownikiem a procesorem: sterownik ma rejestry do pamiętania danych i sygnałów sterujących, które czyta lub pisze procesor. Realizacja komunikacji: 1. specjalne polecenia wejścia-wyjścia - określają przesył danych na adres portu. 2. odwzorowanie operacji wejścia-wyjścia w pamięci operacyjnej (memory-mapped I/O).
Adapter główny (host adapter) - sterownik jako oddzielna płyta z układami elektronicznymi.

Rejestry portu wejścia-wyjścia:

- stan (status) – do czytania przez CPU. Mogą oznaczać zakończenie bieżącego polecenia, dostępność bajta do czytania w rejestrze danych wejściowych, wykrycie błędu w urządzeniu.
- sterowanie (control) – do zapisywania przez CPU w celu rozpoczęcia poleceń, zmiany trybu pracy.
- dane wyjściowe (data-out) – do zapisywania przez CPU w celu wysłania ich do urządzenia.
- dane wejściowe (data-in) – do czytania przez CPU w celu pobrania informacji z urządzenia.

W sterownikach mogą występować układy FIFO w celu zwiększenia pojemności dla danych wejściowych lub wyjściowych.

Odpytywanie (polling)

Uzgadnianie (handshaking) w schemacie producent-konsument:

Urządzenie ma swój bit zajętości (busy): 1 - urządzenie zajęte, 0 - gotowe do przyjęcia polecenia. Procesor ma bit gotowości polecenia (command-ready) w rejestrze poleceń: 1 - polecenie jest dostępne dla urządzenia.

- Procesor czeka, aż bit zajętości będzie miał wartość 0 - jest to czekanie aktywne, czyli odpytywanie (czyta nieustannie rejestr stanu).
- Procesor przygotowuje słowo danych, ustawia bit gotowości.
- Sterownik urządzenia, zauważywszy bit ustawiony bit gotowości, ustawia bit zajętości, rozpoczyna się operacja wejścia-wyjścia.
- Sterownik czyści bit gotowości oraz bit błędy w rejestrze stanu, następnie czyści bit zajętości.

Odpytywanie nie jest wydajne, jeśli sterownik i urządzenie nie są szybkie - procesor nie wykonuje innych czynności.

Przerwania (interrupt)

Linia zgłaszania przerwań (interrupt request line) - procesor bada ją po wykonaniu każdego rozkazu. Gdy wykryje, że sterownik coś zasygnalizował, rozpoczyna się

procedura obsługi przerwania (interrupt-handler). Zostaje rozpoznana przyczyna przerwania, wykonuje się odpowiednie czynności i następuje polecenie powrotu z przerwania, przywracające procesor do stanu sprzed jego zgłoszenia.

Optymalizacja obsługi przerwań:

- Maskowanie - w procesorze występują dwie linie zgłaszania przerwań. Jedna z nich jest dla przerwań niemaskowanych - zarezerwowanych np. dla nieusuwalnych błędów pamięci. Druga jest dla przerwań maskowalnych - używanych przez sterowniki urządzeń, wyłącza się ją przed wykonaniem krytycznych ciągów instrukcji.
- Wektor przerwań - zawiera adresy procedur obsługi przerwań, dzięki niemu zredukowane jest przeszukiwanie wszystkich możliwych źródeł przerwań, aby dowiedzieć się, które należy obsłużyć. Zazwyczaj urządzeń jest więcej niż pozycji w wektorze. Wtedy stosuje się łańcuch przerwań - elementy wektora wskazują na czoło listy procedur obsługi przerwań.
- System poziomów priorytetów przerwań (interrupt priority level) - umożliwia opóźnianie obsługi procedur o niższym priorytecie bez maskowania wszystkich przerwań.

Przerwania mogą służyć też do obsługi sytuacji wyjątkowych, stronicowania pamięci wirtualnej (przy braku strony), zarządzania przebiegiem sterowania w jądrze.

Bezpośredni dostęp do pamięci

Wykorzystywane w przypadku urządzeń transmitujących wielkie ilości informacji w celu uniknięcia stosowana programowego wejścia-wyjścia (PIO). Stosowane przy użyciu sterownika bezpośredniego dostępu do pamięci (DMA - direct memory access).

Aby przesyłać w trybie DMA, procesor główny zapisuje w pamięci blok sterujący DMA, który zawiera wskaźnik do źródła przesyłania i miejsca docelowego oraz liczbę bajtów do przesłania.

Aby rozpocząć przesyłanie, CPU zapisuje adres bloku sterującego w sterowniku DMA, który wtedy przejmuje nadzór nad szyną pamięci, aby przesyłanie odbywało się bez pomocy CPU.

Przesyłanie między DMA a sterownikiem urządzenia:

- sterownik ustawia sygnał zamówienia DMA (DMA request) gdy jest gotowy do przesłania danych.
- sterownik DMA przejmuje szynę pamięci, tworzy adres, ustawia sygnał w potwierdzeniu DMA (DMA acknowledge).
- odbiór sygnały potwierdzenia DMA przez sterownik urządzenia -> przesyła słowa danych, usunięcie sygnały zamówienia DMA.

Podczas przesyłania w trybie DMA procesor nie ma dostępu do pamięci operacyjnej, może korzystać z pamięci tylko z pamięci podręcznej.

Użytkowy interfejs wejścia-wyjścia

Dostęp do urządzeń opiera się na interfejsie - daje to możliwość jednolitego traktowania urządzeń, korzystanie z nowych rodzajów bez naruszania systemu operacyjnego. Między podsystemem wejścia-wyjścia w jądrze a sterownikami urządzeń znajduje się warstwa modułów urządzeń, która ukrywa różnice pomiędzy sterownikami urządzeń. Wywołania systemowe wejścia-wyjścia obudowują działanie urządzeń w ogólne klasy, ukrywające różnice sprzętowe przed aplikacjami.

Różnice między urządzeniami wejścia-wyjścia:

- tryb przesyłania danych: znakowy (pojedyncze bajty przesyłane po kolei) - terminal / blokowy (cały blok bajtów zostaje przesłany na raz) - dysk.

- sposób dostępu: sekwencyjny (uporządkowany sposób przesyłania danych) - modem / swobodny (przesył danych z dowolnego miejsca ich przechowywania) - CD-ROM.

- organizacja przesyłania: synchroniczna (przesłanie w przewidywalnym z góry czasie) - taśma / asynchroniczna (nieregularne i nieprzewidywalne czasy odpowiedzi) - klawiatura.

- dzielenie: użytkowanie wspólne (urządzenie może być dzielone współbieżnie przez wiele procesów lub wątków) - dysk / na zasadzie wyłączności - dedykowane (tylko jeden wątek/proces może z niego korzystać w danym czasie) - taśma.

- szybkość działania: różna ze względu na czas odnajdywania danych, przesyłania, opóźnienia między operacjami czy rotacją nośnika danych.

- kierunek przesyłania: tylko czytanie - CD-ROM / tylko pisanie - sterownik graficzny / czytanie i pisanie - dysk.

Funkcja ioctl (w systemie UNIX) - umożliwia aplikacji dostęp do dowolnej możliwości zaimplementowanej przez jakikolwiek moduł sterujący, bez konieczności obmyślania nowego odwołania do systemu.

Urządzenia blokowe i znakowe:

Interfejs strumienia znaków: polecenia get i put dla pojedynczych znaków. Oprogramowanie biblioteczne może dawać możliwość przesyłania całych wierszy tekstu i ich redagowania. Urządzenia: myszka, klawiatura, modem.

Interfejs urządzeń blokowych: polecenia czytaj, pisz, szukaj (urządzenia o dostępie bezpośrednim). Dostęp do urządzenia przez interfejs systemu plików bądź dostęp surowy - traktowanie urządzenia jak liniową tablicę bloków.

Urządzenia sieciowe:

Z racji licznych różnic między nimi a we-wy dyskowym mają one osobny interfejs. Dzięki interfejsowi gniazd aplikacja może utworzyć gniazdo i połączyć je ze zdalnym adresem, co daje możliwość połączenia i przesyłu danych. Interfejs ten ma funkcję select, która umożliwia sprawdzenie, które z gniazd może przyjąć nowy pakiet danych.

Zegary, czasomierze:

Funkcje: podawanie bieżącego czasu, upływającego czasu, powodowanie wykonania danej operacji w zadanej chwili.

Czasomierz programowalny (programmable interval timer) - można ustawić go na pewien okres, po którym powoduje przerwanie. Może to być jednorazowe lub powtarzać się i okresowo generować przerwania.

Wejście-wyjście z blokowanie oraz bez blokowania:

Blokowanie - zawieszenie procesu, aplikacja trafia na kolejkę procesów czekających, po zakończeniu operacji wejścia wyjścia wraca na kolejkę procesów gotowych. Powszechnie stosowane z racji łatwości użycia i zrozumienia.

Bez blokowania - w przypadku buforowania wejścia-wyjścia. Można zaimplementować wielowątkowo. Wywołanie nieblokowane nie wstrzymuje działania

aplikacji, kończy się tak szybko jak to tylko możliwe i daje aplikacji informacje o liczbie przesłanych bajtów (czy pełna zamówiona liczba, czy mniej).

Asynchroniczne odwołanie - proces nie czeka na zakończenie operacji wejścia-wyjścia, aplikacja kontynuuje swoje działanie, informacja o zakończeniu operacji we-wy zostaje podesłana aplikacji przez podsystem wejścia-wyjścia.

Podsystem wejścia-wyjścia w jądrze

Buforowanie:

Bufor (buffer) – obszar w pamięci, gdzie przechowuje się dane przesyłane między dwoma urządzeniami / między urządzeniem a aplikacją. Powody stosowania bufora:

- radzenie sobie z różnicą szybkości działania urządzeń,
- dopasowanie urządzeń o różnych rozmiarach przesyłanych jednostek danych,
- semantyka kopii – gwarancja, że niezależnie od zmian w buforze aplikacji, na dysku będzie wersja z chwili odwołania się przez aplikację do systemu.

Przechowywanie podręczne:

W pamięci podręcznej przechowujemy kopie danych – dostęp do nich jest znacznie szybszy niż do oryginałów, co zwiększa wydajność.

Różnica między buforem: bufor może zawierać jedyną istniejącą kopię danych, przechowywanie podręczne to utrzymywanie w szybkiej pamięci kopii danych, które są zapisane gdzie indziej.

Spooling:

Spooling – użycie bufora do przechowywania danych przeznaczonych dla urządzenia, które nie dopuszcza do przeplatania danych w przeznaczonym dla niego strumieniu. Przykładem takie urządzenia jest drukarka – nie można dopuścić, aby dane przeznaczone do drukowania zostały między sobą pomieszane.

Rezerwowanie urządzeń:

Przydział urządzeń na wyłączność przez system operacyjny. Do tego celu są wywołania systemowe przydziału i zwalniania urządzenia. Trzeba uważać na zakleszczenia.

Obsługa błędów:

Systemowe wywołanie we-wy zwraca bit informujący czy operacja zakończyła się sukcesem czy niepowodzeniem. Przy awarii zwracany jest kod błędu, mówiący o jego charakterze (errno – służąca do tego zmienna w SO UNIX). Raporty o błędach znajdują się w dziennikach systemowych.

Struktury danych jądra:

Obsługa różnic operacji we-wy za pomocą metod obiektowych oraz przekazywania komunikatów – zamówienie jest przekształcane na komunikat wysyłany za pośrednictwem jądra do zarządcy wejścia-wyjścia.

Ochrona wejścia-wyjścia

Operacje we-wy mają status operacji uprzywilejowanych, muszą odbywać się za pomocą wywołań systemowych. Muszą być chronione miejsce odwzorowań we-wy w pamięci operacyjnej oraz porty.

Przekształcanie zamówień wejścia-wyjścia na operacje sprzętowe

Przykładowy proces czytania pliku z dysku:

- określenie na którym urządzeniu znajduje się plik,
- przetłumaczenie nazwy na jej reprezentację na urządzeniu,
- operacja czytania danych z dysku do bufora,
- udostępnienie danych procesowi, który je zamówił,
- zwrócenie sterowania do procesu.

Strumienie

Strumienie (streams) – mechanizm umożliwiający dynamiczne zestawianie kodu modułów obsługi we-wy w potoki. Strumień to półdupleksowy kanał komunikacyjny między procesem poziomu użytkownika a urządzeniem. Składa się z: interfejsów czoła (nagłówka) strumienia do procesu użytkowego, zakończeń modułu sterującego łączących z urządzeniem, modułów strumienia między nimi. Każdy moduł zawiera kolejkę czytania i kolejkę pisania. Strumieni można używać do komunikacji międzyprocesowej i sieciowej.

Wydajność

Operacje wejście-wyjście stanowią główny czynnik sprawności systemu (przerwania, kopiowanie danych, przełączanie kontekstu, wykonywanie kodu we-wy, obsługi modułów w jądrze).

Poprawa wydajności:

- mniejsza liczba przełączeń kontekstu,
- mniej kopiowania danych w pamięci podczas przekazu urządzenie - aplikacja,
- mniejsza częstotliwość występowania przerwań – stosowanie wielkich przesyłań, przemyślane sterowniki, odpytywanie,
- korzystanie z DMA – zwiększenie współbieżności,
- równoważenie wydajności procesora, pamięci, szyny i operacji we-wy.