Asynchroniczne porty szeregowe

- Wiele urządzeń podłącza się przez porty szeregowe RS-232C w trybie asynchronicznym.
- Unix posiada standardowy driver portu szeregowego z dobrze opracowanym interfacem programowym i administracyjnym.
- Dawniej większość użytkowników włączała się przez terminale asynchroniczne, co wymagało obsługi portu szeregowego w tym celu (program `getty`). Dziś używa się tak tylko modemów.
- Modemy bywają także użwane do nawiązywania połączeń wychodzących, a czasami zarówno przychodzących jak i wychodzących.
- Szeregowych portów używają również niektóre drukarki.
- Szeregowa, asynchroniczna transmisja danych wymaga ustawienia wielu parametrów, i w podstawowej wersji zawiera tylko jednostronne sterowanie przepływem danych (wstrzymywanie transmisji gdy odebrano nadmiar danych).
- Transmisja szeregowa jest zwykle wolniejsza niż inne rodzaje połączeń (np. Ethernet), i długość łączy jest ograniczona (ograniczenie ma związek z predkością).

Sygnały RS-232

nr	dte-dce	symb	nazwa sygnału	nr	dte-dce	symb	nazwa sygnału
1		fg	**frame ground**	14	$\longrightarrow$	std	secondary
2	$\longrightarrow$	td	**transmit data**				transmitted data
3	$\longleftarrow$	rd	**receive data**	15	$\longleftarrow$	tc	transmitted clock
4	$\longrightarrow$	rts	**request to send**	16	$\longleftarrow$	srd	secondary received data
5	$\longleftarrow$	cts	**clear to send**	17	$\longleftarrow$	rc	received clock
6	$\longleftarrow$	dsr	**data set ready**	18		(na)	(not assigned)
7		sg	**signal ground**	19	$\longrightarrow$	srts	secondary request
8	$\longleftarrow$	dcd	**data carrier detect**				to send
9		-	positive test voltage	20	$\longrightarrow$	dtr	**data terminal ready**
10		-	negative test voltage	21	$\longrightarrow$	sq	signal quality detector
11		(na)	(not assigned)	22	$\longleftarrow$	ri	**ring indicator**
12	$\longleftarrow$	sdcd	secondary data	23	$\longleftrightarrow$	drs	data rate selector
			carrier detect	24	$\longrightarrow$	scte	serial clock
13	$\longleftarrow$	scts	secondary clear				transmit external
			to send	25		busy	busy

Elementy asynchronicznej transmisji szeregowej

- standard RS-232C obejmuje połączenie między portem komputera lub urządzenia peryferyjnego (DTE — *Data Terminal Equipment*) a modemem telekomunikacyjnym (DCE — *Data Communications Equipment*)
- pozwala na jednoczesną dwukierunkową transmisję szeregową (*full-duplex*), z możliwością wstrzymywania przesyłania danych przez DCE, oraz automatycznego nawiązywania i odbierania połączeń przez linie telefoniczne (sygnał DCD)
- parametry: prędkość, parzystość, format ramki (liczba bitów w bajcie, dodatkowe bity na początku i na końcu transmitowanego bajtu)
- sterowanie przepływem (wstrzymywanie transmisji przez stronę odbierającą): standard przewiduje sterowanie sprzętowe tylko od strony DCE (obecnie wiele urządzeń stosuje je również od strony DTE), istnieje również pseudo-standard sterowania programowego (XON/XOFF)
- bezpośrednie łączenie DTE z DTE: (*null-modem*)
- ograniczenie długości linii: 50 stóp

Konfiguracja portu modemowego

1. Podłącz terminal lub modem do portu DCE-DTE kablem (w przypadku terminala lokalnego może być konieczny specjalny kabel DTE-DTE).
2. Wybierz parametry transmisji szeregowej; znajdź lub utwórz właściwy opis w pliku `/etc/gettydefs`
3. Znajdź w systemie pliki specjalne drivera portu szeregowego, i wpisz parametry wywołania `getty` do pliku konfiguracyjnego `/etc/inittab`
4. Wymuś zastosowanie przez init nowej konfiguracji (`kill -HUP 1` lub `telinit -q`).
5. Ustal typ terminala i zainstaluj opis terminfo lub termcap.
6. Zaprogramuj parametry modemu (w jego pamięci trwałej). Ogólnie, modemy nastawia się na tryb „małomówny” i „chętnie zrywający połączenia”. Niektóre nowe wersje programu getty komunikują się z modemami w trybie „gadatliwym” (Linux).
7. Dla skonfigurowania jednoczesnego inicjowania i przyjmowania połączeń przez modem zmodyfikuj odpowiednio ustawienia `getty`
8. Skonfiguruj właściwe oprogramowanie połączeń wychodzących (uucp, slip, ppp, itp.).

System drukowania Unixa

W Unixie istnieje system drukowania pochodzący z czasów drukarek mechanicznych, umieszczonych w centralnej sali (hali) komputerowej i obsługiwanych przez operatora, których możliwości konfiguracji i opcje drukowania sprowadzały się do:

- obsługi podstawowych znaków sterujących: TAB, CR, LF, FF,
- zakładanie różnego rodzaju papieru (np. kilkuwarstwowego), ciągniętego na rolkach przez perforację,
- zakładanie kółek lub głowic drukujących z różnymi zestawami znaków.

W systemie tym głównym problemem było rozróżnienie i rozdzielenie wydruków należących do różnych użytkowników, i z drugiej strony uniknięciu czekania gdy inni użytkownicy akurat wysłali duże zadania do drukowania zajmujące drukarkę np. przez następne 2 godziny. Dla rozwiązania pierwszego problemu system drukowania ma opcję (często domyślnie włączoną i trudną do wyłączenia) drukowania strony lub kilku stron rozdzielających, tzw. *banner page*.

Ten system przetrwał w niezmienionej postaci, pomimo ogromnego rozwoju technologii drukarek i ich wielu możliwości i opcji, głównie ze względu na:

- szybki i niestabilny rozwój formatów reprezentacji grafiki i formatów opisu strony,
- pojawienie się standardowego formatu opisu strony Postscriptu, który wbudowany w hardware drukarek powoduje, że wystarczy, by programy generowały dane do drukowania w tym formacie, a system drukowania może pozostać nieświadomy jakie dane i w jakim formacie są drukowane.

Dlatego Postscript jest mniej lub bardziej niejawnym standardem danych drukowanych w systamch Unixa, a administracja systemem drukowania polega na konfiguracji portów, nazw drukarek, uprawnień użytkowników do korzystania z nich, i zaledwie kilku subtelnych drobiazgów.

Elementy systemu drukowania AT&T/BSD

W wariantach AT&T i BSD Unixa istnieją odmienne i niekompatybilne (choć podobne) systemy drukowania. Oba składają się z następujących elementów:

- *spooler* — program obsługujący porty drukarek i kolejki zadań drukowania, zapewnia ustawienia parametrów portu, retransmisje, itp. (AT&T:`lpsched`/BSD:`lpd`)

- programy użytkownika przekazujące żądania drukowania spoolerowi (AT&T:`lp`/BSD:`lpr`), sprawdzające zawartość kolejki drukowania i stan zadań (AT&T:`lpstat`/BSD:`lpq`), usuwające zadania z kolejki (AT&T:`cancel`/BSD:`lprm`), i inne

- narzędzia administracyjne (AT&T: `lpadmin`, `accept`, `reject`, `enable`, `disable`, i wiele innych, oraz BSD: `lpc`)

- pliki konfiguracyjne:
  AT&T: `/usr/spool/lp/model/*`
  BSD: `/etc/printcap`

Główną różnicą między tymi systemami była większa prostota, ale jednocześnie możliwość obsługi drukarek sieciowych systemu BSD, i większa złożoność z jednoczesnym brakiem możliwości sieciowej systemu AT&T. Stąd w prawie wszystkich systemach drukowania sieciowego stosowany jest obecnie protokół BSD, tzw. *lpd*. Natomiast w większości współczesnych Unixów istnieje połączenie obu tych systemów, dający dualny interface do jednego zestawu drukarek.

Przykładowy plik /etc/printcap

```
# Copyright (c) 1983 Regents of the University of California.
# All rights reserved.
#
# @(#)etc.printcap 5.2 (Berkeley) 5/5/88
#
# This file was generated by /usr/sbin/magicfilterconfig.

lp|lab010|Panasonic KX-P1180:\
        :lp=/dev/lp0:sd=/var/spool/lpd/lab010:\
        :sh:pw#80:pl#72:px#1440:mx#0:\
        :if=/etc/magicfilter/epson9-filter:\
        :af=/var/log/lp-acct:lf=/var/log/lp-errs:
hp5|Hp5|masza:\
        :lp=:sd=/var/spool/lpd/hp5:rm=156.17.9.3:rp=hp5:lpr_bounce:\
        :sh:pw#80:pl#72:px#1440:mx#0:\
        :af=/var/log/lp-acct:lf=/var/log/lp-errs:
hp3|Hp3|wiera|bepa:\
        :lp=:sd=/var/spool/lpd/hp3:rm=156.17.9.23:rp=lp:lpr_bounce:\
        :sh:pw#80:pl#72:px#1440:mx#0:\
        :af=/var/log/lp-acct:lf=/var/log/lp-errs:
```

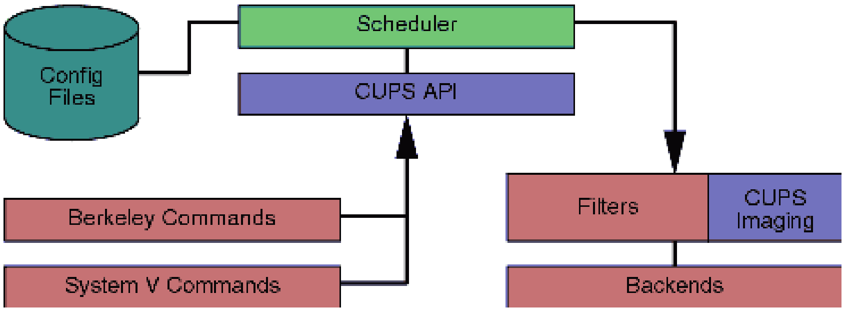

Filtry systemu drukowania

W obu systemach drukowania istnieje możliwość skonfigurowania filtrów, czyli programów, które przetwarzają strumień danych przesyłanych przez spooler na drukarkę, co pozwala na przykład na:

- automatyczną obsługę wydruku na drukarkach obsługujących mniej lub bardziej specyficzne języki opisu strony lub grafiki, jak np.: PCL, HPGL, itp.,
- tworzenie drukarek wirtualnych przyjmujących dane w dowolnych formatach stosowanych w danej instytucji, przez zaprogramowanie filtrów przekształcających te formaty na formaty/języki obsługiwane przez posiadane drukarki,
- wykrywanie błędów użytkowników w przesyłaniu plików na drukarki, np. sprawdzanie czy plik przesyłany na daną drukarkę ma poprawną postać i może zostać poprawnie wydrukowany, np. czy użytkownik nie wysłał 10 MB-owego programu binarnego na drukarkę tekstową lub postscriptową.

CUPS — nowy system drukowania?

Publicznie dostępny system drukowania:



Założenia:

- scheduler oparty na protokole HTTP
- rozpoznawanie typów plików oparte na schemacie MIME
- obsługa dowolnego typu drukarek (pliki PPD + filtry)
- obsługa dowolnego typu interface'u drukarki (szeregowe, równoległe, USB, sieciowe) przez „wtyczki" backend
  na przykład istnieje backend oparty na systemie samba pozwalający na drukowanie na drukarkach podłączonych do Windowsa

- interface do systemów drukowania ATT i BSD: lp/lpr i lpstat/lpc