

1 Drucken unter Linux

1.1 Wie funktioniert das Drucken unter Linux

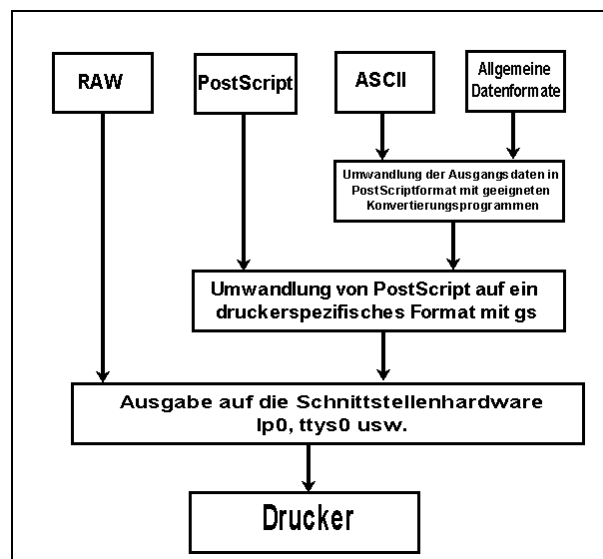
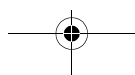


Abb. 1.1: Schema des Druckvorgangs unter Linux

Das Drucken unter Linux ist auf den ersten Blick ein komplizierter und verwirrender Prozess. Bei der Verarbeitung einer Druckdatei sind mehrere Programme, Skripte und Filter beteiligt.

Prinzipiell gliedert sich das Drucken in drei Schritte:

1. Umwandlung einer Druckdatei, automatisch oder manuell, eines gegebenen Formats – GIF, Latex (dvi), ASCII usw. – in das PostScript-Format. Dies übernehmen die Skripte *asfilter* bei SuSE bzw. *magicfilter*





Wie funktioniert das Drucken unter Linux

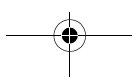
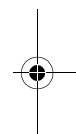
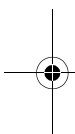
- bei Red Hat. Bei einer Installation des Drucksystems LPD (beim alternativen Drucksystem CUPS ist die Organisation anders, siehe Kapitel 4). Diese Filter rufen ggf. weitere Konvertierungsprogramme auf.
2. Umwandlung der PostScript-Daten in das druckerspezifische Format. Diese Aufgabe übernehmen Device-Treiber von *gs* (*Ghostsript*) bei dem LDP-Drucksystem bzw. die Filter *rastertoplc/rastertoprinter* bei einer CUPS-Installation. Mehr zu diesem Programm und seinen Parametern findet man in den Kapiteln 2 und 4.
 3. Ausgabe dieser Daten über die eingestellte Hardware-Schnittstelle an den Drucker.

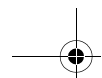
Die Verbindung zwischen den drei oberen Lagen und die Parameterübergabe werden von den Spooling- oder Druckersystemen gemanaget. Die drei gebräuchlichsten Druckersysteme sind CUPS, LPD und LPRng.

Spooling-Systeme:

Linux ist wie sein „großer Bruder“ Unix von Haus aus ein Multiuser-System, d.h. es können gleichzeitig dezentral mehrere User auf demselben System eingeloggt sein und arbeiten. Dabei kann sich ein solches Rechner-system auch über ein ganzes Gebäude erstrecken.

Spooling-Systeme haben die Aufgabe, Druckaufträge (-jobs) von allen Eingabegeräten (Terminals, PCs über Netzwerkverbindung) anzunehmen und an die entsprechenden, zumeist zentral stehenden Ausgabegeräte, auch Plotter, weiterzuleiten. Zusätzlich zu dieser simplen Weiterleitung muss aber noch einige Verwaltungsarbeit geleistet werden. Zunächst müssen die Druckjobs gemäß ihrer Priorität in der richtigen Reihenfolge zwischengespeichert werden. Es muss sichergestellt werden, dass bei einem Systemausfall die Druckjobs, besonders wenn sie erst zum Teil ausgeführt wurden, nicht verloren gehen. Dazu werden vornehmlich dem Administrator verschiedene Werkzeuge an die Hand gegeben, um den Status des Spooling-Systems abzurufen. Zudem können Drucker freigeschaltet oder gesperrt bzw. Druckjobs angehalten, wieder gestartet, gelöscht, auf andere Drucker umgeleitet oder die Druckreihenfolge geändert werden. Außerdem besteht auf dieser Ebene die Möglichkeit, alle Druckaktivitäten zu protokollieren.





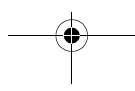
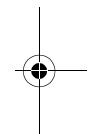
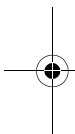
Drucken unter Linux

Spooling-Systeme geben auf der anderen Seite allen Usern die Möglichkeit, unkompliziert von jedem beliebigen Arbeitsplatz aus, auf jeden installierten Drucker (Plotter) zuzugreifen.

1.2 Warum kann das Drucken unter Linux problematisch sein

Der Vorteil, den ein dezentrales und frei konfigurierbares Drucksystem in Verbindung mit der Forderung einer guten Netzwerkanbindung von Druckern und Druckerservern bietet, beinhaltet auch eine große Anzahl von Fehlerquellen. Zudem erschweren die Struktur und die Ausführung mit ihren verschiedenen Programmen, die sich teils überlappen, teils ausschließen oder auch stören, ein klares Verständnis des Zusammenwirkens der einzelnen Systemteile.

Diese Probleme sind überwiegend historisch bedingt und basieren auf gewachsenen Strukturen und Standards. Unix, das Vorbild von Linux, ist von seiner Konzeption her ein netzwerkorientiertes, dezentrales Multiuser-Betriebssystem. Da es zudem ein wissenschaftliches System war, das an den Universitäten entwickelt wurde, hat man sehr auf plattformübergreifende Standards geachtet. Aus diesem Grunde existieren viele Anwendungen als Skripte, die von jedem System, ganz gleich wie die Hardwareausführung ist, interpretiert werden können. Diese Skripte zur Erweiterung von Betriebssystemmöglichkeiten wurden z.T. nach Absprache von verschiedenen Personen erstellt. Aus dieser Anfangszeit stammt auch die Übereinkunft von Unix/Linux, das PostScript-Format als Ausgabestandard zu definieren. Dateien, Messergebnisse, Veröffentlichungen und Arbeitspapiere konnten in diesem allgemeinen Format an jedes andere System verschickt werden, um dort mit den Standards vor Ort ohne Qualitätsverlust ausgegeben zu werden. Auf die spezifischen Hardwaregegebenheiten des Zielsystems musste dabei keine Rücksicht genommen werden. Dies ist letztlich der Grund, warum es unter Linux so viele Skripte und kleine Programme gibt, die die verschiedenen Dienste starten und konfigurieren. Das gilt insbesondere für das Drucksystem.





Die Aufgabe des Druckertreibers

Der Vorteil, den diese Modularität und das einheitliche Ausgabeformat bieten – man muss bei Fehlerbeseitigung und Updates nicht das gesamte System erneuern –, kehrt sich aber ins Gegenteil um, wenn es Probleme beim Drucken gibt.

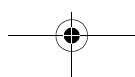
Zum einen existieren heute mehr mögliche Datenformate sowohl auf der Eingangsseite z.B. für Textdateien, Tabellen und Bilder, als auch auf der Ausgabeseite, denn die PostScript-Drucker sind auch heutzutage eher im höheren Preissegment und somit im professionellen Bereich anzutreffen. Die dadurch notwendigen Konvertierungen der Daten bringen naturgemäß auch eine gewisse Anzahl von Konfigurationsmöglichkeiten und somit Fehlerquellen mit sich.

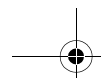
Zum anderen ist es nicht einfach, die Übersicht zu behalten, welche Skripte und welche Programme an dem Druckprozess beteiligt sind und wo die Fehlerquellen liegen könnten. Oft befindet sich der Fehler nicht im Skript selbst, sondern in einer fehlerhaften Parameterübergabe bzw. in fehlenden Programmteilen wie z.B. Fontdateien. Endgültig verwirrend kann es werden, wenn die Programmteile wie Fonts oder spezielle Datenfilter zwar vorhanden sind, aber im falschen Verzeichnis stehen und somit nicht gefunden werden können. Die Linux-Drucksysteme haben dann nämlich die unangenehme Eigenschaft, den Druckprozess ohne Fehlermeldung und ohne Ausdruck abzubrechen bzw. Fehlerprotokolle an einem unbekanntem Ort abzuspeichern. Dies gilt in verstärktem Maße für die grafischen Benutzeroberflächen *KDE* und *Gnome*, bei denen die Abwicklung des Druckprozesses in den Hintergrund verlagert wird.

Die Schwierigkeiten, die beim Drucken unter Linux auftreten können, resultieren also aus dem Zusammenspiel der verschiedenen Komponenten des jeweiligen, installierten Drucksystems.

1.3 Die Aufgabe des Druckertreibers

Moderne Drucker und zwar sowohl PostScript-Drucker als auch Laserdrucker und Tintenstrahldrucker setzen eine Druckseite als „Bild“ im Speicher aus einzelnen Punkten zusammen, auch dann, wenn lediglich Text gedruckt





Drucken unter Linux

werden soll. Dieses „Bild“ wird dann unter Berücksichtigung der jeweiligen druckerspezifischen Steuerkommandos an das Ausgabe-Device lokal an die serielle, parallele bzw. USB-Schnittstelle oder über eine Netzwerkverbindung weitergeleitet.

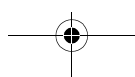
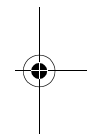
Die Hauptaufgabe des Druckertreibers ist demnach, einen Algorithmus zur Rasterung der Druckdatei bereitzustellen, den Raster-Image-Prozess (RIP). Dabei muss der RIP alle Formate, von reinem ASCII-Text über PostScript-Format bis zu allen gängigen Bildformaten, verarbeiten können.

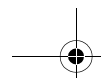
Bei PostScript-Druckern steckt diese Intelligenz in der Druckerhardware selbst, bei den anderen Druckern wird der Rasterprozess vom Druckertreiber übernommen.

Linux bietet von Haus aus nur eine Grundfunktionalität zur Ausgabe von PostScript-Dateien auf einen lokalen Drucker. Hier setzen die Spooling-Systeme an.

1.4 Was sind Drucksysteme?

Mit dem Begriff „Drucksystem“ ist bei Linux und allen anderen Multiuser-Betriebssystemen eine mehr oder weniger lose Kombination von Werkzeugen, Filtern und Treibern gemeint, mit deren Hilfe hauptsächlich die Benutzer- und Druck-Job-Verwaltung (Spooling) bewerkstelligt wird. Für die konkrete Ansteuerung eines modernen Druckers auf Basis einer der verbreiteten Druckersprachen wie etwa PCL kommt bei Linux das Programm Ghostscript zum Einsatz, weil die weitaus meisten Anwenderprogramme unter Linux PostScript-Code ausgeben. Diese Zusammenhänge werden später erläutert. Falls in Verbindung mit dem Drucken unter Linux vom so genannten klassischen Drucksystem LPR (*lpd*) die Rede ist, ist damit in der Regel eine Kombination des BSD-Spooling-Systems der University of California mit dem *lpd*-Drucker-Daemon und einem Filtersystem zur Formaterkennung (in der Regel *apsfilter*) gemeint. Verfügt der Anwender nicht über einen PostScript-Drucker, wird das klassische Drucksystem normalerweise noch um den eigentlichen Druckertreiber zur Erzeugung des Rasterbildes für die Ausgabeeinheit des Druckers auf Basis von Ghostscript





Was sind Drucksysteme?

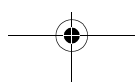
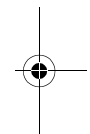
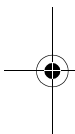
erweitert. Dieses klassische Drucksystem ist aber in die Jahre gekommen und weist einige deutliche Mängel und Sicherheitslücken auf, sodass bereits seit einiger Zeit Alternativen zur Verfügung stehen. Dazu gehören z. B. der Nachfolger des klassischen Drucksystems LPRng oder die Drucksysteme LPQ und CUPS. Das moderne Drucksystem CUPS werden wir uns in diesem Zusammenhang noch detaillierter ansehen.

1.4.1 Welche Drucksysteme gibt es?

Im Gegensatz zu Windows gibt es unter Linux diverse Drucksysteme, wobei zu jedem System auch ein eigenes Spooling-System gehört.

Das bekannteste Drucksystem ist zugleich auch das älteste: LPD/LPR, das von der Universität Berkeley entwickelt wurde. Aufgrund der historischen Gegebenheiten und der Weiterentwicklungen besonders in der Netzwerktechnik, entspricht dieses Drucksystem nicht mehr dem Stand der Technik und wird auch nicht mehr bei allen Distributionen als Standardsystem installiert.

Seit ein paar Jahren erobert ein anderes Drucksystem den Linuxmarkt: CUPS (Common Unix Printing System). Die Stärken dieses Systems liegen nicht so sehr in einer Steigerung der Druckqualität, sondern in der vollständigen Umsetzung des IPP-Standards, sodass CUPS ideal für große, netzwerkorientierte und dezentrale Multiuser-Systeme ist. Dem Anwender fällt außerdem die nahtlose Integration des Drucksystems in die KDE-Desktop-Oberfläche auf. Zudem ermöglichen die offene Struktur der verwendeten PPD-Dateien – sie können mit jedem Editor gelesen und verändert werden – und der Software-Raster-Engine Ghostscript die Einbindung von nicht PostScript-Druckern. Dieses Feature ist besonders für Besitzer von GDI-Druckern von Interesse, da hier Druckertreiber für nicht unterstützte Drucker von privaten Entwicklern, wie sie sich auf der Homepage <http://www.linuxprinting.org> zusammen gefunden haben, erstellt werden können.





Drucken unter Linux

Der jüngste Vertreter in der Riege der Drucksysteme ist LPRng (Line Printing next generation). Wie der Name schon vermuten lässt, ist dies die Weiterentwicklung des betagten LPD/LPR-Systems. Bei dieser Variante wurden die interne Struktur überarbeitet und die Einbindung von Netzwerkdruckern neu implementiert. LPRng wird von allen Distributionen als Drucksystem bei der Installation mit angeboten.

Neben diesen drei Hauptvertretern von Drucksystemen gibt es noch einige andere Systeme wie PDQ und SysV, wobei hier der Vollständigkeit halber lediglich deren Existenz erwähnt werden soll.

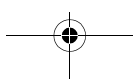
1.4.2 Drucksystem LPD/LPR

Das Drucksystem LPD/LPR, im Original BSD Line-Printer-Daemon, ist das älteste Drucksystem in der Unix/Linux-Welt. Dabei ist die Bezeichnung LPD etwas verwirrend, da sie sowohl für das Drucksystem als auch für den zentralen Druckdienst einer Multiuser-Umgebung verwendet wird. Dieser Dienst nimmt die einzelnen Druckaufträge der Benutzer entgegen und weist sie den zugehörigen Warteschlangen und Filtern zu.

Das Drucksystem LPD bezeichnet dagegen die Spoolerkomponente des BSD-Drucksystems. Dazu gehört eine spezielle Client/Server-Architektur, die beim Netzwerkdruck zum Einsatz kommt. Wie bei den anderen Drucksystemen lässt sich LPR nicht auf ein zentrales Programm eingrenzen, was die Arbeit etwas kompliziert macht. Die Hauptkomponenten sind der Filter *apsfilter*, der den Ablauf des Druckvorgangs steuert und die Software-Raster-Engine Ghostscript.

1.4.3 Drucksystem CUPS

CUPS benutzt zum Ansteuern eines Druckers die von Adobe entwickelte PPD-Struktur. Dabei verarbeitet CUPS neben den eigenen PPD-Dateien sowohl die originalen PPD-Dateien von Adobe als auch die Pseudo-PPD-Dateien, die von dem Linux-Printing-Projekt (siehe auch <http://www.linuxprinting.org>) für fast alle Nicht-PostScript-Drucker, insbesondere für



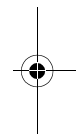
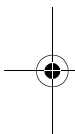


Was sind Drucksysteme?

GDI-Drucker, bereitgestellt werden. Allerdings kann es beim Betrieb von PostScript-Druckern mit originalen Adobe PPD-Dateien Probleme geben, nämlich dann, wenn die PPD-Datei neben PostScript- und druckerspezifischen Befehlen auch Aufrufe an andere Programme wie DLLs enthält. Ein Hauptmerkmal von CUPS ist die vollständige Umsetzung des IPP-Protokolls, was gerade die Administration von Druckern über ein Netzwerk bei Mehrplatzsystemen erheblich vereinfacht.

Die erzielbare Druckqualität ist allerdings bei CUPS nur dann besser, wenn bei der Rasterbild-Erzeugung (RIP) die mitgelieferten PPD-Dateien benutzt werden, da nur dann der CUPS-eigene Raster Engine benutzt wird und nicht Ghostscript.

CUPS bietet die Möglichkeit Druckerklassen einzurichten, d.h. man kann mehrere Drucker in einer Warteschlange zusammenfassen. Dadurch ist es möglich auch größere Druckerpools zu bilden, die dann zentral verwaltet werden können. Alle Druckaufträge, die an eine solche Klasse gesendet werden, werden automatisch an den ersten freien Drucker weitergeleitet. So können Wartezeiten beim Drucken verkürzt und eine gleichmäßige Auslastung der Druckressourcen erreicht werden. LPRng wurde speziell für sehr große, dezentrale, netzwerkorientierte Drucksysteme konzipiert, sodass auch mehrere Hundert Drucker verwaltet werden können.

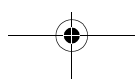


1.4.4 Drucksystem LPRng

Das Drucksystem LPRng (Line Printer next generation) ist weitgehend kompatibel zu dem alten LPD-System, d.h. die dort existierenden Befehle gelten auch hier, obwohl die einzelnen Systemteile komplett neu entwickelt wurden.

Mit dieser Neuimplementierung gehen einige Verbesserungen einher. LPRng ist sicherer als sein Vorgänger und erleichtert das Einrichten von Druckern speziell für Netzwerke.

LPRng besitzt genau wie CUPS die Möglichkeit, Druckerklassen einzurichten. Zudem ist LPRng noch hervorragend dokumentiert, was es nicht zuletzt auch für den normalen Benutzer interessant macht.





Drucken unter Linux

Wegen der Kompatibilität zum LPD-System stehen auch weiterhin die Programme *lpr*, *lpq*, *lprm* und *lpc* mit ihren Optionen zur Verfügung (siehe auch Kapitel 1.4.5). Hinzu kommt das Programm *lpstat*. Es liefert dem User Informationen über die definierten Druckerklassen, die einzelnen Drucker und den Status der Druckjobs.

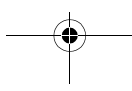
Nachfolgend die wichtigsten Optionen für *lpstat*:

- a: zeigt für alle Drucker an, ob sie bereit sind, Druckaufträge anzunehmen
- c: zeigt alle Klassen an
- d: zeigt den aktuellen Standarddrucker an
- r: zeigt an, ob der CUPS-Server läuft oder nicht
- R: zeigt die Rangfolge der Druckjobs an
- s: zeigt eine Statusübersicht an (Defaultdrucker, Liste aller Druckerklassen und Drucker inklusive Device-Name usw.)
- t: zeigt alle verfügbaren Informationen an; ist eine Zusammenfassung der Optionen -r, -d, -c, -d, -v, -a, -p, und -o
- v [*Warteschlange(n)*]: zeigt die Drucker und das zugewiesene Schnittstellendevice an. Wenn keine Warteschlange angegeben wird, werden die Schnittstellen für alle Warteschlangen ausgegeben.

Konfiguriert wird das System über die folgenden Dateien in ihrem Defaultinstallationspfad:

Verzeichnis	Beschreibung
/etc/printcap	Druckerbeschreibungs-Datei
/etc/lpd.conf	LPRng Konfigurationsdatei
/etc/lpd.perms	Druckerberechtigungen
/var/spool/printer*	spool-Verzeichnis
/var/spool/printer*/printer	Lock-File der Warteschlange
/var/spool/printer*/control.printer	Warteschlangen-Kontrolldatei

Tab. 1.1: Die LPRng-Konfigurationsdateien





Was sind Drucksysteme?

Verzeichnis	Beschreibung
/var/spool/printer*/active.printer	Aktiver Drucker
/var/spool/printer*/log.printer	Log-Datei

Tab. 1.1: Die LPRng-Konfigurationsdateien (Fortsetzung)

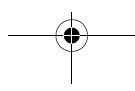
1.4.5 Der Linux-Druckdaemon und seine Kommandozeilen-Befehle

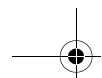
Der *lpd* (engl. line printer daemon) wird bei allen Drucksystemen mit installiert. Somit steht immer unabhängig vom gewählten Drucksystem ein Standardbefehlssatz für die Konsole zur Verfügung, mit dem man die Druckerwarteschlangen administrieren und verwalten kann. Aus diesem Grund werden der *lpd* und seine Unterprogramme hier behandelt, bevor auf die verschiedenen Drucksysteme LPD, LPRng und CUPS im Einzelnen eingegangen wird.

Der *lpd* wird beim Systemstart aktiviert und läuft als Daemon im Hintergrund. Bei der SuSE-Distribution wird er durch das Skript `/sbin/init.d/lpd` gestartet, wenn in der `/etc/rc.config` die Variable `START_LPD=yes` gesetzt wurde.

Der *lpd*-Daemon stellt beim Start anhand der `/etc/printcap` fest, welche Druckerwarteschlangen definiert sind. Das gilt auch für CUPS, das die Datei `/etc/printcap` eigentlich nicht mehr benötigt. Sie wird von CUPS trotzdem angelegt, aber nur mit den Namen der angelegten Drucker gefüllt. Die Aufgabe von *lpd* ist es, die Ausführung der gespoolten Aufträge (engl. Jobs) zu organisieren. Im Einzelnen bedeutet dies:

- Management der lokalen Warteschlangen: Er schickt die Druckdaten eines jeden Jobs durch den passenden Filter (festgelegt durch den Eintrag der Warteschlange in `/etc/printcap` sowie durch explizite Angaben in der Job-Steuerdatei) und dann zur Druckerschnittstelle.
- Berücksichtigung der Reihenfolge der Jobs in den Druckerwarteschlangen.





Drucken unter Linux

- Überwachung des Status der Warteschlangen und Drucker (Dateistatus in den lokalen Spoolverzeichnissen); gibt auf Verlangen Auskunft darüber.
- Weiterleitung der Druckaufträge an Warteschlangen auf entfernte Rechner an den dortigen *lpd*-Daemon.
- Annahme von Druckaufträgen von entfernten Rechnern für die eigene lokale Warteschlange oder deren Abweisung bei ungenügender Autorisierung.

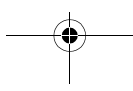
Während die letzten Punkte automatisch im Hintergrund ablaufen, stellt der Druckerdaemon dem Administrator gerade für die erste Aufgabe einige Tools zur Verfügung.

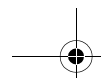
-lpr: Mit dem Kommando *lpr -Pname printfile* wird die Datei *printfile* an die Druckerwarteschlange *name* geschickt. Von ihr wird sie in dem Spool-Verzeichnis */var/spool/lpd/Warteschlange* oder in ein von */etc/printcap* festgelegtes Verzeichnis, siehe Kapitel 3.1, kopiert. Danach wird sie, wenn der Drucker bereit und diese Datei an der Reihe ist, ausgedruckt. Nachdem der Druckauftrag komplett an den Drucker geschickt wurde, wird er aus der Druckerwarteschlange entfernt. Diese Vorgehensweise stellt sicher, dass keine Druckjobs verloren gehen, wenn das System während des Druckens heruntergefahren wird oder abstürzt.

In der Konsole ist der Befehl *lpr* aber nicht nur für den Ausdruck von Dateien zuständig, sondern kann auch zum Drucken von Ausgaben auf den *stout*-Kanal also auf die Konsole selbst benutzt werden. So leitet z.B. der Befehl *ls -l | lpr* die Ausgabe der Directory direkt auf den Drucker um.

-lpq: Das Kommando *lpq -P<name>* zeigt eigene Druckjobs in der Warteschlange *name* an. Der Befehl *lpq -P<name>* ergibt die folgende Ausgabe *name is ready and printing*

Rank	Owner	Job	Files	Total Size
<i>active</i>	<i>tux</i>	<i>676</i>	<i>Hallo.txt</i>	<i>259420 bytes</i>
<i>1st</i>	<i>tux</i>	<i>677</i>	<i>brief.dvi</i>	<i>11578 bytes</i>
<i>2nd</i>	<i>tux</i>	<i>683</i>	<i>bild.gif</i>	<i>bytes</i>





Was sind Drucksysteme?

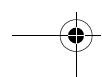
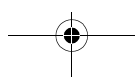
-lprm: Das Kommando *lprm* löscht Druckaufträge aus einer Warteschlange, sofern der Druckauftrag dem Benutzer gehört, der *lprm* aufruft. Der Administrator kann alle Druckjobs löschen. Zum Löschen muss man die Warteschlange und die Job-ID angeben.

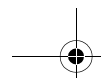
-lpc: Dieses Kommando dient der Steuerung der Warteschlangen. Anstelle des Warteschlangen-Namens kann dabei auch *all* (= alle) angegeben werden. Fehlt die Angabe *<warteschlange>*, so wirkt der Aufruf wie *<all>*, d.h. ein Statusbericht für alle Warteschlangen wird ausgegeben.

Die wichtigsten Optionen sind:

Option	Beschreibung
status warteschlange	gibt einen Statusbericht, z.B. <i>lex:</i> <i>printer is on device 'parallel' speed -1</i> <i>queuing is enabled</i> <i>printing is enabled</i> <i>no entries</i> <i>daemon present</i>
disable warteschlange	stoppt die Aufnahme neuer Jobs in die Warteschlange
enable warteschlange	gibt die Warteschlange für die Aufnahme neuer Jobs frei
stop warteschlange	stoppt das Ausdrucken von Jobs aus der Warteschlange; der gerade im Druck befindliche Job wird noch beendet
start warteschlange	nimmt das Ausdrucken von Jobs aus der Warteschlange wieder auf
down warteschlange	wirkt wie die Optionen <i>disable</i> und <i>stop</i> zusammen

Tab. 1.2: Optionen für das *lpc*-Kommando



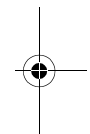
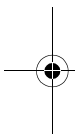


Drucken unter Linux

Option	Beschreibung
up warteschlange	hat dieselbe Wirkung wie die Optionen enable und start zusammen
abort warteschlange	ist identisch zu down, nur dass ein gerade im Druck befindlicher Job sofort abgebrochen wird. Die Jobs bleiben erhalten und können nach einem Restart der Warteschlange (up) weiter bearbeitet werden

Tab. 1.2: Optionen für das *lpc*-Kommando (Fortsetzung)

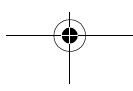
Man kann diese Kommandos mit dem *lpc*-Befehl gleich in der Kommandozeile eingeben (z.B. *lpc status*) oder *lpc* ohne Parameter aufrufen: Dann wird ein Dialogmodus mit der Eingabeaufforderung *lpc>* (engl. Prompt) gestartet, der die Eingabe von *lpc*-Kommandos erwartet. Mit *quit* oder *exit* wird dieser Dialog beendet.



1.5 Der Rasterprozess

Das primäre Ziel sämtlicher Raster-Technologien besteht immer darin, Grau- oder Farbstufen zu simulieren. Die einzelnen Raster-Algorithmen wandeln dabei die Graustufen in eine Kombination von Bildpunkten (half-tone dots) um, etwa wenn das Endgerät, die so genannte „marking engine“ des Druckers, nur schwarze und weiße Bildpunkte kennt, beispielsweise bei Laserdruckern. Unter einer „marking engine“ versteht man in diesem Zusammenhang denjenigen Teil des Ausgabegerätes, der dafür zuständig ist, die Rasterpunkte auf das Papier aufzubringen.

Bei PostScript-Druckern wird die von der Anwendung generierte PostScript-Datei zusammen mit den eingestellten Optionen wie Wahl des Papierschachts, Blattgröße oder doppelseitiger Druck direkt übergeben. Der Drucker bereitet dann das geräteunabhängige PostScript-Format mit Hilfe seines eingebauten RIP's (Raster Image Process[or]) eigenständig auf, d.h.





Der Rasterprozess

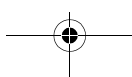
er erzeugt ein Rasterformat, das an die eigene „marking engine“ weitergeleitet wird.

Bei Nicht-PostScript-Druckern, insbesondere Tintenstrahldruckern, gestaltet sich der Rasterprozess ungleich aufwändiger. Tintenstrahldrucker müssen die im Bild vorgegebenen kontinuierlichen Werte von cyan, magenta, gelb und schwarz gemäß des gewählten Algorithmus durch diskrete Werte ersetzen. Außerdem kann die Rasterung nicht im Drucker geschehen, sondern muss vom Treiber auf dem angeschlossenen Rechner vorgenommen werden.

1.5.1 Die Rastertechnologie

Um bei Zeilendruckern einen möglichst streifenfreien Ausdruck zu erreichen, kommen dazu verschiedene herstellereigene Rastertechnologien wie Microweaving (Epson) oder RET-III (Hewlett Packard) zum Einsatz. In diesem Zusammenhang haben sich herstellereigene Druckersprachen etabliert. Zu den bekanntesten gehören Adobe's PostScript, PCL (Print Control Language) von HP oder ESC/P von Epson. GDI (Graphic Device Interface) von Microsoft fällt hier etwas aus dem Rahmen, da es keine einheitliche Druckersprache ist. Die Anforderungen für die Treiberalgorithmen sind immens. Bei Auflösungen von 1440 dpi mit 4 Farben bzw. 6 beim Fotodruck entsteht ein erheblicher Berechnungsaufwand zur Erzeugung der von der marking engine eines modernen Druckers benötigten Rasterbilddaten. Aktuelle Tintenstrahldrucker-Modelle verwenden je Farbe bis zu 320 einzelne Düsen (z. B. Canon S630, 320 Düsen für schwarz, 256 Düsen pro Farbe). Bei Mittelklassemodellen, wie etwa dem Epson Stylus C70, sind Druckköpfe mit ca. 180 Düsen für schwarz und ca. 60 Düsen für jede Farbe üblich.

Beim Herstellungsprozess der Druckköpfe sind gewisse mechanische Beschränkungen zu beachten, die nicht unterschritten werden können. So kann der vertikale Abstand der Tintendüsen nicht beliebig klein gemacht werden, da zwei aus derselben Düse stammende Zeilen niemals direkt untereinander positioniert werden können. Eine Technik, um einen möglichst streifenfreien Ausdruck zu bekommen, ist den Ausdruck zu „weben“.



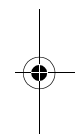


Drucken unter Linux

Tintenstrahldrucker besitzen im Gegensatz zu PostScript-Druckern keinerlei Eigenintelligenz, deshalb muss die Zuordnung der Düsen zu den Reihen durch den verwendeten Druckertreiber erfolgen. Diese Vorgehensweise erfordert eine enorme Rechenleistung des Rechners. Noch höher sind allerdings die Anforderungen an den Rechner beim fotorealistischen Druck, der mit 6 bzw. 7 Farben durchgeführt wird.

Die Druckerhersteller HP und Canon arbeiten mit variablen Tröpfchengrößen, C-Ret III – Technologie bzw. Droplet genannt. Das Überlappen der Druckzeilen bei Epson heißt Microweaving.

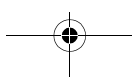
Geht es beim Weaving primär um die geeignete Wahl der „Abfolge“ einzelner Zeilen, die der Druckkopf zu Papier bringt, so besteht bei der C-Ret III- und der Droplet-Technologie die Hauptaufgabe des dazugehörigen Algorithmus darin, zu ermitteln, ob an einer bestimmten Stelle überhaupt ein Tintentropfen zu versprühen ist oder nicht, und wenn ja, welche Farbmischung und welche Intensität. Alles in allem ist daher die Berechnung des endgültigen Punktmusters für das Ausgabegerät aus den PostScript-Daten der Anwendung, gegebenenfalls unter Einbeziehung eines Farbkorrektur-Systems, ein komplexer Vorgang.



1.5.2 Die Farbprofile

Beim fotorealistischen Druck muss der Rasteralgorithmus Millionen unterschiedlicher Farbnuancen des Originals in eine Druckvorlage umsetzen können. Alle Hersteller arbeiten hierbei mit ähnlichen Grundprinzipien, doch alle basieren auf einer „Rasterung“ der Druckfarben, wobei die Farben jeweils in einem regelmäßigen (*ordere*) oder unregelmäßigen Muster (*Error-Diffusion*) aufgetragen werden.

Um eine möglichst farbechte Wiedergabe beim Drucken zu erzielen, besteht die Hauptaufgabe des Rasteralgorithmus darin, aus den Rot-, Grün- und Blau-Komponenten der Bildschirmfarben die notwendigen Intensitäten der Drucker-Komplementärfarben zu berechnen. Dabei ist zu beachten, dass jeder Druckerhersteller unterschiedliche Farbnuancen bei seinen Grundfarben benutzt, die in der Regel nie exakt den Komplementärfarben der Bildschirmfarben entsprechen. Damit ist es für den Druckertreiber (Rasterprozessor) prinzipiell nicht möglich, ein reines Blau etwa durch das





Was sind Druckersprachen?

Mischungsverhältnis von 50% Magenta und 50% Cyan zu erzeugen, denn je nach Farbstick der Druckergrundfarben müssen die Farbanteile unterschiedlich gewichtet werden.

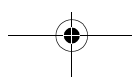
Die Farbkorrektur ist ebenfalls erforderlich, weil immer mit einer leichten Verunreinigung der Druckerfarben (z.B. kleiner Gelb-Anteil im Cyan) zu rechnen ist. Außerdem kann das verwendete Papier – Umweltpapier, rein weißes Papier, Glossie-Paper, Folie oder Fotopapier – die Druckfarben je nach aufgetragener Farbmenge unterschiedlich stark ineinander fließen lassen. Zudem verändert die chemische Beschichtung der unterschiedlichen Papiere jeden Farbton geringfügig.

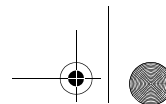
Insgesamt gestaltet sich also eine geeignete „Beeinflussung“ der resultierenden Farbe durch Tinte (Farbton, Fließigenschaften, Mischungseigenschaften etc.) und Papier (chemische Beschichtung, Absorptionsverhalten, Grundfarbe etc.) sehr schwierig, sodass die im Idealfall aufzutragende Tintenmenge auch mit einer komplizierten mathematischen Formel gar nicht oder nur näherungsweise aus den Bildschirmfarben errechenbar ist.

Aufgrund dieser komplexen Beziehungen zwischen Druckerhardware, Tinteneigenschaften und Papiereigenschaften kann in der Praxis ein Druckertreiber einen wirklich realistischen Ausdruck nur erzielen, wenn er für jede Bildschirmfarbe und jede Papiersorte ein Farbprofil anlegt, das dann auch individuell vom Anwender angepasst werden kann. Ein solches Farbprofil muss mittels diverser Testdrucke auf unterschiedlichen Papiersorten mit verschiedenen Farbmustern erstellt werden. Das Drucksystem CUPS bzw. das kprinter-Modul der Desktop-Oberfläche KDE 3.0 ermöglichen eine Erstellung und Einrichtung eines solchen Farbprofils und der so genannten Gamma-Korrektur (siehe Kapitel 5.2).

1.6 Was sind Druckersprachen?

Druckersprachen sind, vereinfacht gesagt, binäre Anweisungen, die über eine Schnittstelle an den Drucker ausgegeben, von diesem interpretiert und dann in Bewegungen des Druckkopfes umgesetzt werden. Dabei unterscheiden sie sich sehr in ihrer Komplexität und ihren Möglichkeiten. Die Typenraddrucker und Nadeldrucker der ersten Generation akzeptierten im





Drucken unter Linux

Großen und Ganzen den ASCII-Code, der auch schon bei den Fernschreibmaschinen verwendet wurde. Hinzu kamen noch einige rudimentäre Steuerzeichen. Damit war es jedem Benutzer möglich, den Drucker direkt aus Programmen anzusprechen und in bestimmte Modi zu versetzen, z.B. einen bestimmten implementierten Zeichensatz einzuschalten oder in den Grafikmodus zu wechseln.

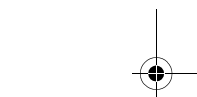
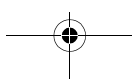
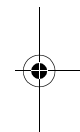
Je komplexer die Druckersprache ist, desto mehr Möglichkeiten beinhaltet sie auch, detaillierte Einstellungen für den Ausdruck vorzunehmen. Bei Highlevel-Sprachen wie PostScript können auch sehr individuelle Angaben zur Druckkopfpositionierung und zur Farbmischung durch Mehrfachdruck einer Zeile vorgenommen werden. Der Grad der Komplexität von Druckersprachen reicht von kryptischen Steuersequenzen wie bei ESC-P bis hin zu vektororientierten Seitenbeschreibungssprachen wie PostScript.

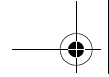
1.6.1 PostScript

Wie so Vieles in der Linux-Architektur stammt der PostScript-Standard auch aus der Unix-Welt. PostScript ist eigentlich eine Skriptsprache zum Beschreiben von Seiteninhalten – und zwar unabhängig vom Ausgabegerät – die 1985 von Adobe eingeführt wurde.

In der Anfangszeit der Großrechnersysteme waren die Plattformen, d.h. die Rechnerhardware und deren Ausstattung, vollkommen uneinheitlich. Auch gab es keine genormten Ausgabegeräte, obwohl deren Auswahl nicht sehr groß war. Hauptsächlich waren es Typenraddrucker, Nadeldrucker, Plotter, Fotoplotter, Bildschirme und Terminals. Deshalb wurde eine allgemeingültige Sprache geschaffen, die den Inhalt einer Seite und deren Geometrie beschreibt. Die Umsetzung und die Ausgabe waren dann wieder von System zu System unterschiedlich und eine lokale Aufgabe. Im günstigsten Fall gab es für das System einen PostScript-Drucker oder -Plotter; dann war die Ausgabe kein Problem.

Diese Beschränkung auf eine Seitenbeschreibungssprache bot den Vorteil, dass man Texte, Bilder oder Grafiken systemübergreifend mit anderen Unix-Anwendern austauschen konnte. Man erstellte ein Dokument in Post-





Was sind Druckersprachen?

Script und versandte es. Die Anzeige und Ausgabe auf Papier waren die Aufgabe des Systems vor Ort.

Ein weiterer großer Vorteil von PostScript ist, dass es eine vektororientierte Sprache ist, d.h. alle Informationen über Zeichenelemente und Text, deren Größe und Position, liegen als Koordinaten und Vektoren vor. Dadurch kommt es zu keinen Verlusten, wenn man die Datei auf Geräten unterschiedlicher Auflösung und Blattgröße ausgibt. Man kann jedoch sagen, je höher die Auflösung ist, desto schärfer ist die Ausgabe. Aus diesem Grund wird das PostScript-Format auch besonders im Druckgewerbe verwendet.

Diese Vorteile bedingen, dass das PostScript-Format in das spezifische Format des Ausgabegerätes umgewandelt werden muss, bevor es ausgegeben werden kann, was manchmal nicht einfach zu durchschauen ist.

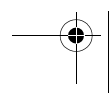
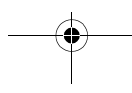
Im Prinzip ähnelt PostScript als Seitenbeschreibungssprache Latex. Zusätzlich ist PostScript aber auch eine vollständige Programmiersprache auf Interpreter-Basis, mit der man Berechnungen, Schleifenoperationen, Fallunterscheidungen usw. programmieren kann. Dieser Teilaspekt von PostScript soll aber hier nur am Rande erwähnt sein. Mehr Informationen kann man unter <http://www.cs.indiana.edu/docproject/programming/postscript/postscript.html> und <http://yoyo.cc.monash.edu.au/~wigs/postscript/> erhalten.

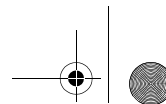
Im Folgenden sollen einige rudimentäre Befehle vorgestellt werden, um dem Leser einen Einblick zu geben, was PostScript ausmacht. Da PostScript eine Interpretersprache ist, braucht man natürlich auch einen Interpreter. Das ist aber kein Problem, da bei allen Linux-Distributionen ein solcher mitgeliefert wird, nämlich das Programm *gs*. Das Programm *gs* wandelt nicht nur die PostScript-Dateien in ein druckerspezifisches Ausgabeformat um (mehr darüber in Kapitel 3), sondern liefert auch eine Interpreterumgebung, wenn man *gs* ohne Parameter aufruft. In der Konsole aufgerufen, öffnet *gs* ein Ausgabefenster, in dem das Ergebnis aller Befehle sofort angezeigt wird.

Zunächst wollen wir einen einfachen Text definieren und dann eine einfache Grafik z.B. ein Quadrat, ausgeben.

Bei jeder Ausgabe müssen zuerst ein paar Basiseinstellungen vorgenommen werden. Für unseren Text wäre wie folgt vorzugehen:

1. Definition, welcher Font in welcher Größe verwendet werden soll.





Drucken unter Linux

2. Definition, wo (an welche Koordinaten) gedruckt werden soll, wobei die untere linke Ecke der Nullpunkt ist.
3. Definition, was gedruckt werden soll.

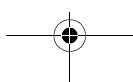
Ein Beispiel könnte sein:

```
/Times-Roman findfont % Font wählen
20 scalefont          % Fontgröße bestimmen 20 dpi
setfont               % als aktuellen Font bestimmen
newpath               % Start mit "new path"-Statement
72 72 moveto          % Textposition angeben
(PostScript is beautiful) show % Text definieren und
    ausgeben
showpage              % Seite darstellen
```

An diesem Beispiel erkennt man auch noch eine Eigenheit von PostScript. Es benutzt die umgekehrte polnische Notation, ähnlich wie sie von den HP Taschenrechnern bekannt ist, d.h. zuerst kommen die Argumente *72 72* als Koordinaten und dann die Operation *moveto* als Kommando, um die Zeichenposition festzulegen.

Zum Zeichnen einer simplen Grafik geht man ähnlich vor:

```
/inch {72 mul} def    % konvertiert inch in dpi (1/72 inch)
newpath               % Start mit new path
1 inch 1 inch moveto  % Anfangspositon anfahren
2 inch 1 inch lineto  % 1. Ecke
2 inch 2 inch lineto  % 2. Ecke
1 inch 2 inch lineto  % 3. Ecke
closepath             % Automatisches Schließen
    der Figur path
stroke                 % Quadrat zeichnen
showpage              % Zeichnung ausgeben
```





Was sind Druckersprachen?

Wenn das Quadrat gefüllt werden soll:

```
/inch {72 mul} def      % konvertiert inch in dpi (1/72 inch)
0.5 setgray            % Grauwert auf 50% setzen
newpath               % Start mit new path
1 inch 1 inch moveto   % Anfangspositon anfahren
2 inch 1 inch lineto   % 1. Ecke
2 inch 2 inch lineto   % 2. Ecke
1 inch 2 inch lineto   % 3. Ecke
closepath             % Automatisches Schließen der Figur
fill                  % Quadrat füllen und zeichnen
showpage              % Zeichnung ausgeben
```

Wie man an den beiden Beispielen sehen kann, müssen die einzelnen Seitenelemente nicht geordnet von oben nach unten eingegeben werden. Wenn man diese Befehle in einer Datei abspeichert (vorzugsweise mit der Endung *.ps), können die Ausgaben mit Ghostscript auch vereinfacht mit

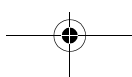
```
gs Datei.ps quit.ps
```

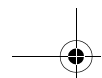
auf dem Bildschirm bzw. auf dem Drucker mit

```
gs -sDEVICE=druckername -sOutputFile=/dev/lp0 Datei.ps quit.ps
```

erfolgen. Die genaue Erklärung dieser Befehlszeilen und weitere Erläuterungen zu Ghostscript finden Sie in Kapitel 2.3.

Da aber nicht alle Daten in der Linux-Welt im PostScript-Format ausgegeben werden, sind noch einige Tools nötig, die die Lücke zwischen den verschiedenen Formaten und PostScript schließen. Diese Tools, die die Umwandlung von allen gängigen Formaten in das PostScript-Format übernehmen, werden in den Kapiteln 3.2.1 und 4.1.5 genauer vorgestellt, wenn die Filter der einzelnen Drucksysteme behandelt werden.





1.6.2 PCL

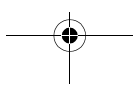
PCL wurde von Hewlett Packard entwickelt und ist in verschiedene Abstraktionslevel unterteilt.

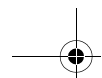
Derzeit gibt es die PCL-Level PCL1 bis PCL5. PCL-Befehle sind kompakte Steueranweisungen, so genannte ESC-Anweisungen, weil alle Kommandos mit dem ESC-Zeichen (Hex 1b) eingeleitet werden, die in den Datenfluss des Druckauftrags eingebettet sind. PCL unterscheidet in dem derzeitigen Aufbau fünf grundlegende Funktionsebenen, die aufeinander aufbauen, wobei PCL5 hinsichtlich der Abstraktionsebene und Funktionsweise mit PostScript vergleichbar ist. Die Funktionalität der einzelnen PCL-Ebenen sind:

PCL-Level	Beschreibung
PCL 1	Print and Space: Ermöglicht ein einfaches Drucken mit Ausgabefunktionen für einzelne Benutzer-Workstations
PCL 2	EDP/Transaction: Allgemeines Drucken für mehrere Benutzer. Erweiterter Satz der PCL 1
PCL 3	Office Word Processing: Dokumenterzeugung in der Textverarbeitung. Erweiterter Satz der PCL 2
PCL 4	Page formatting: Funktionen für den Seitenformatdruck. Erweiterter Satz der PCL 3
PCL 5	Office Publishing: Schriftartenskalierung und HPGL/2 wurden hinzugefügt. Erweiterter Satz der PCL 4

Tab. 1.3: Die PCL-Ebenen

PCL bietet einige Vorteile gegenüber PostScript. Zum einen ist PCL zwei- bis fünfmal schneller als PostScript, insbesondere bei einfachem Text und einfachen Grafiken. Zum anderen benötigt PCL generell weniger Speicherkapazität zum Ausführen eines Druckauftrags. Bei PCL gibt es eine direkte Beziehung zwischen Druckspeicherbedarf und Dateigröße, wodurch eine bessere Skalierbarkeit von Druckerservern möglich wird. Es gibt aber auch bei PCL einige Nachteile. Rein subjektiv scheint es bei einer bestimmten





Was sind Druckersprachen?

Art von Grafiken zu einer schlechteren Druckqualität zu kommen. Ein Großteil solcher qualitativer Unterschiede zwischen PCL und PostScript lässt sich jedoch durch Treiber- oder Anwendungseinstellungen ausgleichen. PostScript-Schriftarten sind im PCL-Modus nicht ohne weiteres verfügbar. Um diese Funktionalität zu nutzen, muss man auf externe Programme zurückgreifen, die PostScript-Fonts in PCL-kompatible Formate umwandeln.

1.6.3 ESC/P

ESC/P ist eine Druckersprache, die von Epson speziell für die eigenen Drucker entwickelt wurde. Diese Entwicklung stammt noch aus der Zeit der Nadeldrucker. Diese wurden mit so genannten Escape-Sequenzen angesteuert. Durch sie konnte schnell mit wenigen Befehlen zwischen verschiedenen im Drucker implementierten Druckfonts oder zwischen Grafik- und Textmodus umgeschaltet werden. Ein Beispiel wäre: *ESC Fontselect FontNr 1 bzw. 27h 51h 01h.*

